Progress®
Corticon®

Corticon
Web Console

Copyright

Last updated with new content: Corticon 7.0

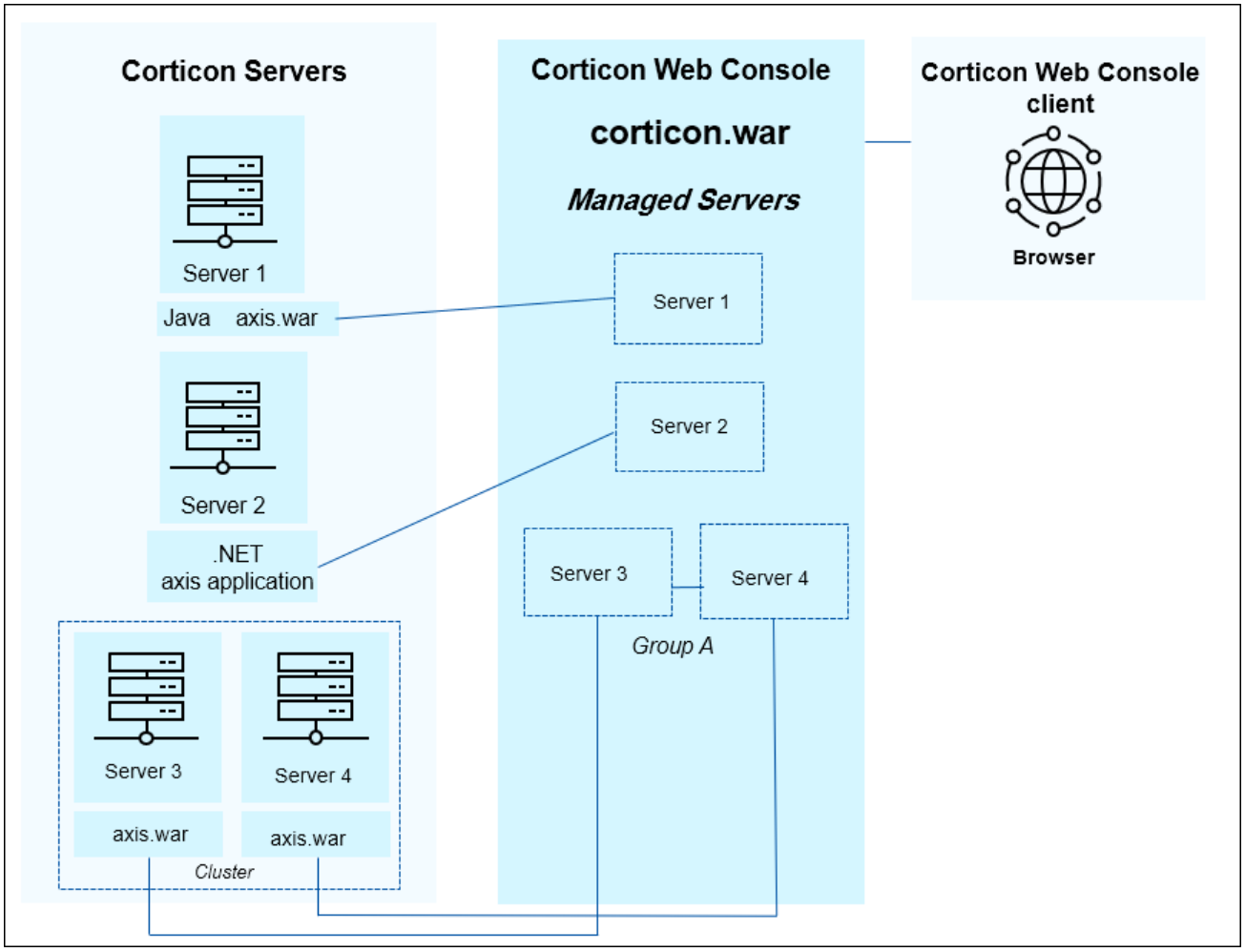
Updated: 2024/01/12

Table of Contents

1

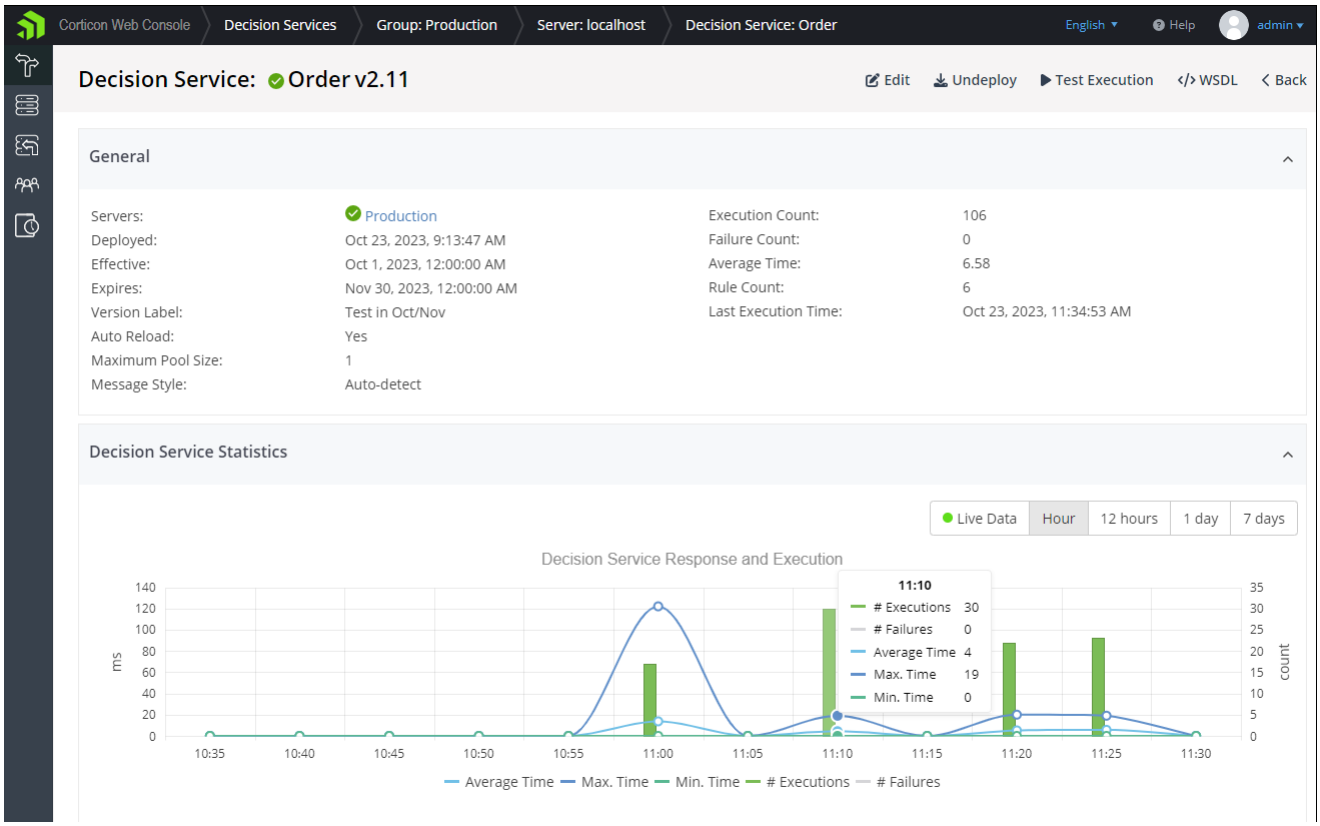
About the Corticon Web Console

Corticon Web Console is a distinct installation option that creates a management server accessed from a browser to manage distributed application servers hosting Corticon deployments, as illustrated:



Corticon's Web Console provides a central point for administering and monitoring your Java and .NET Corticon Decision Services. Through the console you can easily deploy individual Decision Services to one or more Corticon Servers. You can also group related Decision Services into an Application to deploy and manage them as one. Once deployed, you can easily monitor the performance of the Decision Services and Corticon Servers and view both individual and aggregated metrics.

The following image illustrates a Web Console view of a Decision Service with a graph of the responses and executions over a span of a several minutes:



Actions on Decision Services associated with a Server Group are automatically applied to each server member of the group that is running. For example, if you have a Decision Service managed by an Application which is deployed to a Server Group and add another server to the group, the Decision Service will be automatically deployed to the new server. This helps you scale up or scale down the servers in a deployment to meet demand.

The Web Console is a web application that can be installed in the same application server as the Corticon Server for single-server environments or installed separately for multiple-server environments. The choice is yours, depending on the nature of your Corticon deployment. The Web Console maintains configuration information and historical metrics in a local data store. The historical metrics let you see changes in the performance of your Decision Services and Corticon Servers over time.

Corticon's Windows **Start** menu provides shortcut to **Start Corticon Server**. When the Web Console is installed standalone, this starts just the Web Console. When the Web Console is installed together with the Corticon Server, this shortcut starts both of them. Then, the **Corticon Web Console** shortcut launches your default browser to connect to the local Web Console.

This guide describes user activities in the Web Console interface, followed by an administrator's section that touches on architectural features and management functions.

2

User Guide

A server administrator uses a web browser to connect to a running Web Console Server. You will see how the Web Console interface works, how you manage Corticon Servers in a distributed architecture, and how you manage and monitor the Decision Services that run on those servers.

Note: If you are getting started with the Corticon Web Console - Requires access to a running network-accessible Corticon Server that has the Web Console component. See *"How to install Corticon Servers and Web Console" in the Corticon Installation Guide* for installation information.

Note: If you upgraded the Corticon Server - An updated Corticon Serverwith the Web Console component will likely have left residual display and link data in its cache. Clear the browser cache to ensure that the Web Console starts cleanly.

To connect to a running Web Console Server:

- On any device, in a supported browser, enter the hostname where Web Console is running followed by the port value (typically `8850`) and then `/corticon`. For example:

  ```
  http://localhost:8850/corticon
  ```

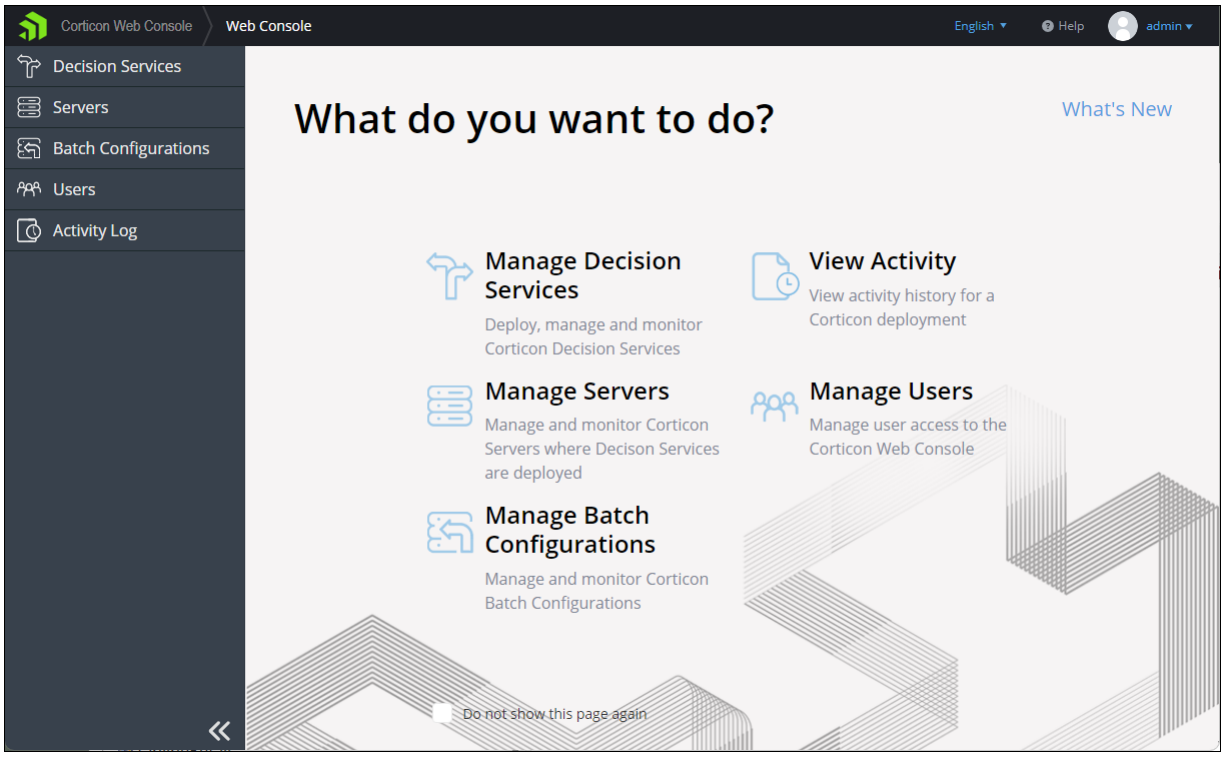
- When you are on the machine that hosts the Web Console installation, simply choose **Start > Progress > CorticonWeb Console**

Logging in to the Web Console

Enter your user credentials in the Web Console login page. When you start using the Web Console, the one pre-defined user is the administrative user, `admin`, with the default password `admin`. If you are the administrator, you should change the default password soon after you log in. Only the `admin` user can add new users. All users have rights to deploy and manage Decision Services. If your role is as a user, obtain your user credentials from your Web Console administrator.

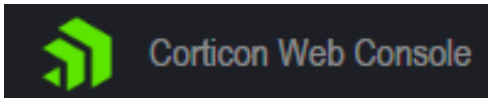
What Do You Want To Do?

When you log in for the first time, you see a welcome page that acquaints you with the Web Console's functions. You also can access **What's New** in Corticon.



Click any action button on the welcome page, the title bar, or the function pane to close the welcome page and open the chosen page.

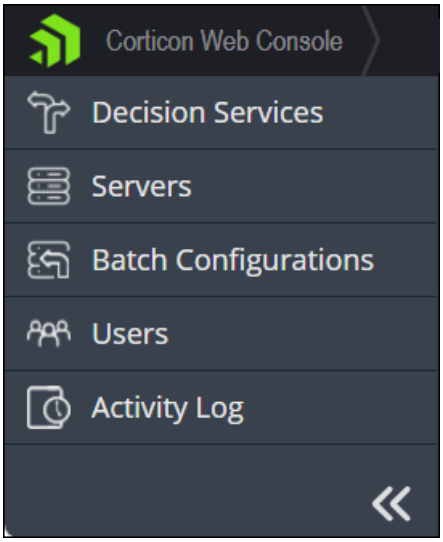
You can re-open the welcome page by clicking in the upper left corner of the page:



Navigation

The general navigation elements of Web Console pages are:

- **Navigation bar** on the left provides access to the functional areas described on the page:



You can choose to toggle the navigation bar display.

- **Title bar:**
  - The navigation path to the current page in the Web Console.
  - **English**: The default language is shown. Choose your preferred available language from its drop-down list to view text in that language as well as localized formatting of dates and numbers.
  - **Help:**
    - **Help Contents:** Opens a new tab linked to the Web Console help for this release.
    - **About**: Version information about the connected Web Console Server.
    - **Community:** Opens a new tab linked to the Progress Corticon community site.
  - ***admin*** (the User Name that enabled log in)
    - **Profile:** Lets the user change their password, full name, and email address.
    - **Logout:** Closes the session and logs the user off the Web Console Server.

Note: Automatic logout - A user gets logged out of their Web Console session when they are inactive for a period of time specified by the Web Console administrator. A warning message is issued several seconds before the Web Console logs out with the opportunity to click **OK** to reset the inactive timeout period.

For details, see the following topics:

- Components in a Corticon deployment
- Decision Services and Applications
- Batch Configurations
- How to view the Activity Log

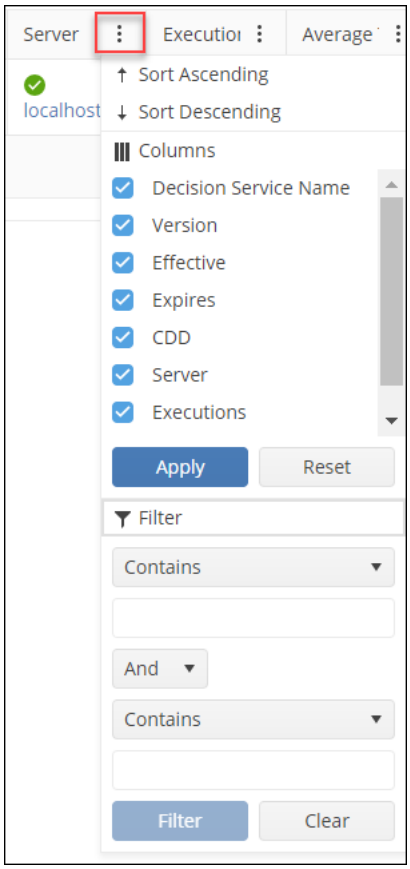
Components in a Corticon deployment

The components that you work with in the Web Console are:

- **Decision Services** - The Corticon Decision Services added to the Web Console. A Decision Service is a set of Corticon rules and supporting assets packaged for deployment.
- **Applications** - Collections of one or more Decision Services to be managed as set. For example, a set of Decision Services in support of a business process that you want to deploy or monitor as a whole.
- **Servers** - Individual instances of Corticon Java or .NET Servers that have been registered with the Web Console. Once registered, the servers are available for deployment of Decision Services.
- **Server Groups** - Groups of one more Servers. Server Groups are useful when you want to deploy Decision Services to a set of Servers. For example, a set of Servers behind a load balancer, or in a regional location.
- **Users** - Defined users who can use the Web Console to administer a Corticon deployment.
- **Activity Logs** - Record of user actions in the Web Console and other asynchronous events such as a server going offline.

Sorting and filtering components

In the pages that list Servers, Decision Services, and Users, you can readily adjust the column sizes and display as well as sort and filter which lines qualify for display by clicking on a column header, as illustrated:



Server groups and Servers

The Web Console allows you to manage and monitor Corticon Servers. The servers can be managed individually or in groups. Server Groups are useful when you want to deploy Decision Services to a set of Corticon Servers. A common use case is a set of Corticon Servers running behind a load balancer where each Server needs to have the same set of Decision Services deployed. Additionally, you can view aggregate metrics for the performance of the servers in a group.

Servers that automatically register with the Web Console

In an elastic deployment environment, new server instances might not be recognized by the Web Console. These new server instances must be added manually to a Server Group in the Web Console. Server instances might spin up and down based on load. When a new server instance can register itself during startup with the Web Console, the Web Console can automatically report the server's metrics.

See Server registration with Web Console for details on the procedures and configuration.

Add Server groups and Servers

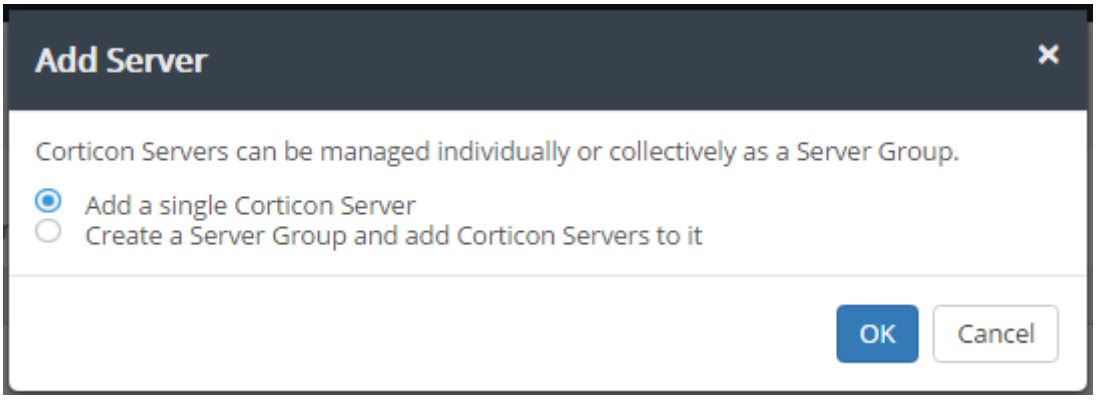
As Corticon Servers are the deployment platform that runs Corticon Decision Services, your Web Console requires that you have one or more Corticon Servers under management so that can you deploy Decision Services and Applications. You can create **Server Groups** to enable common distribution of Decision Services to all servers in the group, and immediate provisioning of new servers added to the group.

Note: When you first start the Web Console in a new installation, no servers are under management unless you installed both Corticon Server for Java and Corticon Web Console. In that case, the Corticon Server is, by default, brought under management in the Web Console as the server `localhost`.

To add servers and server groups:

1. Connect to the Web Console server where you want to add servers and server groups.
2. Click the **Servers** button: 
3. Click **+ Add Server**: 

4. In the **Add Server** dialog box, choose whether to add a single server or a server group:

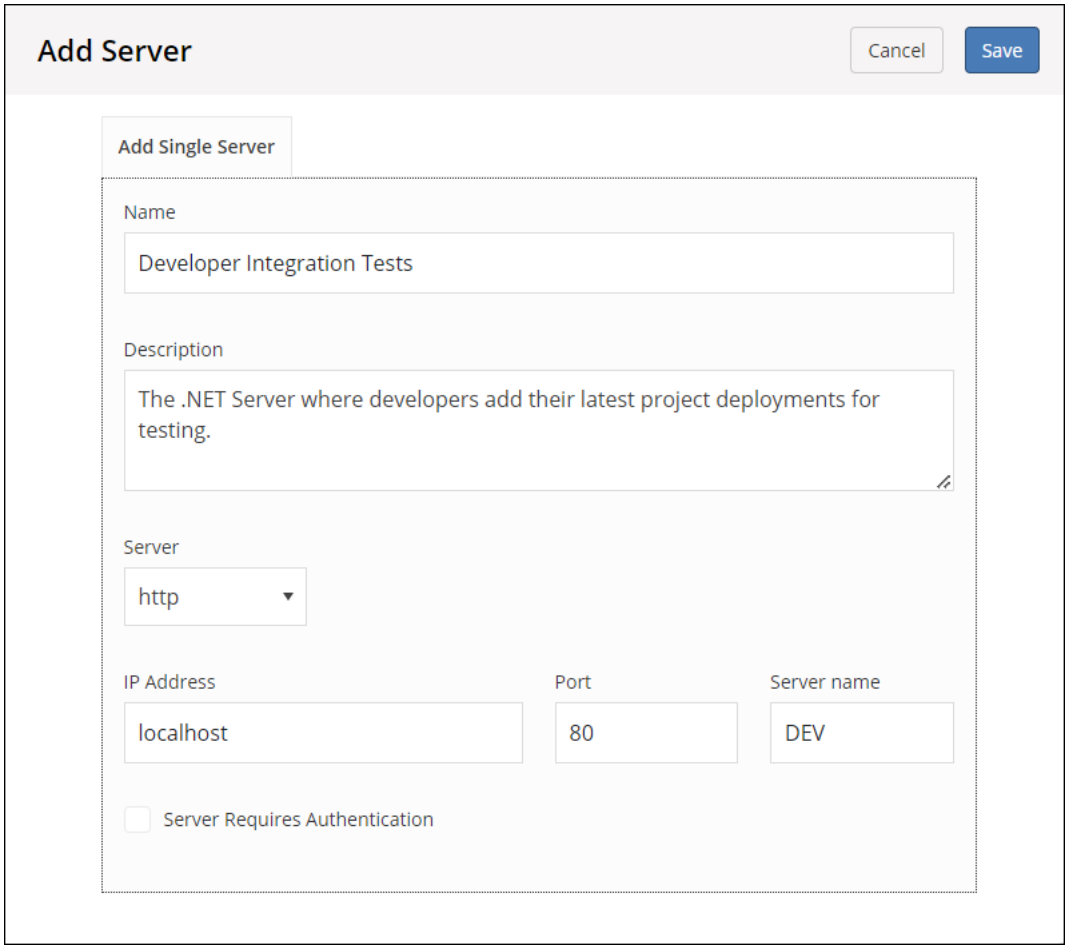


5. Click **OK**.
6. There a few ways to add servers. The following entries are common **Server** information to each of them:
   - **Protocol**: Default is HTTP. You can choose HTTPS, if this server has enabled it.
   - **Hostname**: Enter the DNS-resolvable name or static IP address (avoid *localhost* and *127.0.0.1*)
   - **Port**: 8850 is the default HTTP port, 8851 for HTTPS, 80 typically on IIS
   - **Context URL**: The default is axis
   - **Server Requires Authentication**: When authentication has been enabled on a server, choose this option, and then supply the user name and password for the Web Console to use to establish a connection to the server.

Note: In addition, the default context URL, **axis**, can be replaced with a preferred context URL, such as **CorticonProduction**. This functionality — renaming a default `axis.war` file to a preferred `.war` name — enables multiple server deployments to use the same host port and supporting resources.

Adding a single Corticon Server

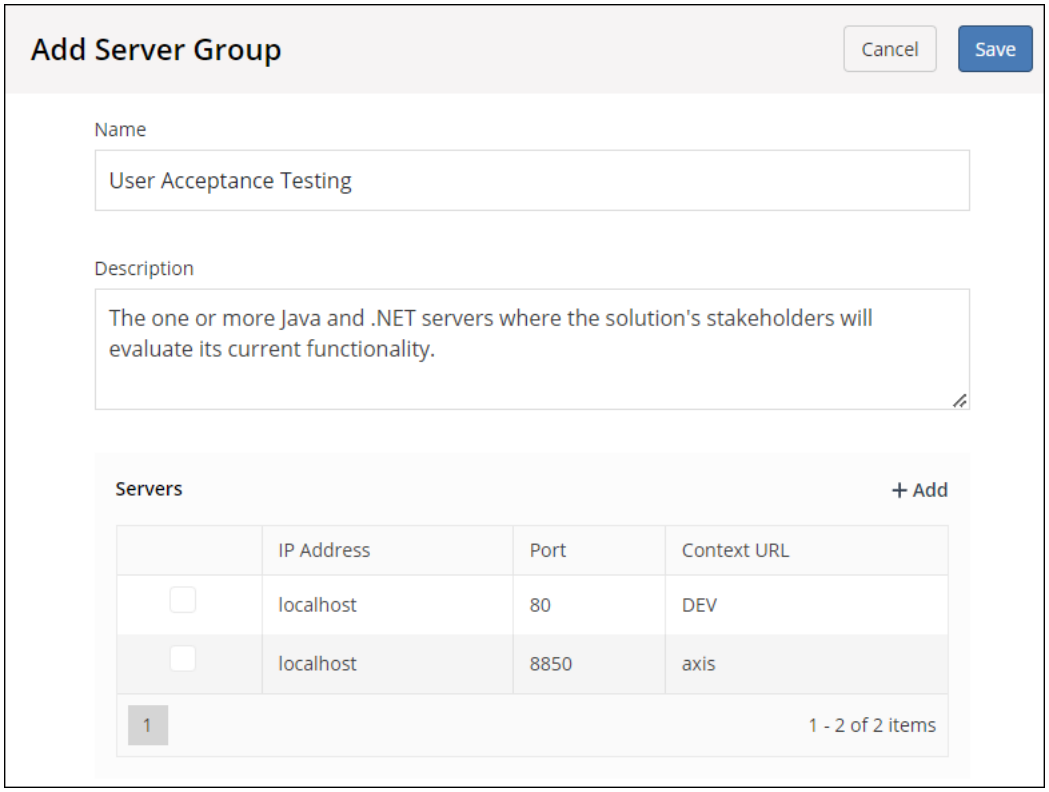
If you choose **Add a single Corticon Server**, the following dialog box opens:



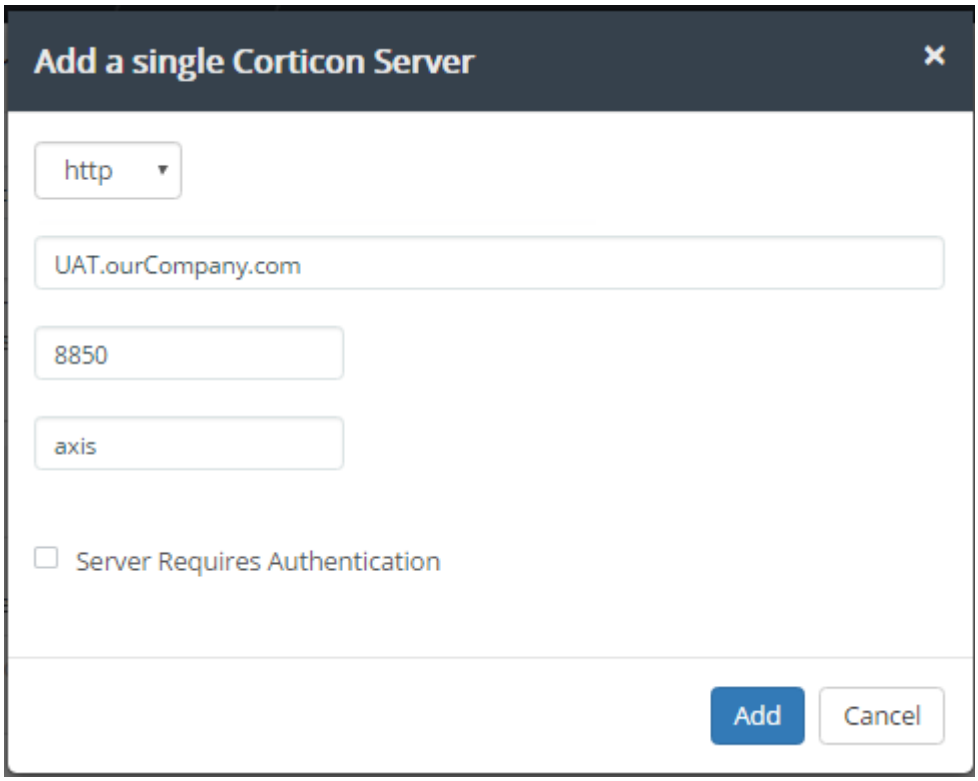
Enter the name you want to describe this server, and a description. Then enter then the server information. Click **Save** when your entries are complete.

Adding a Corticon Server to a new Server Group

If you choose **Create a Server Group and add Servers to it**, the following dialog box opens:



Enter the group name and a description, then click **+ Add** to open the following dialog box:



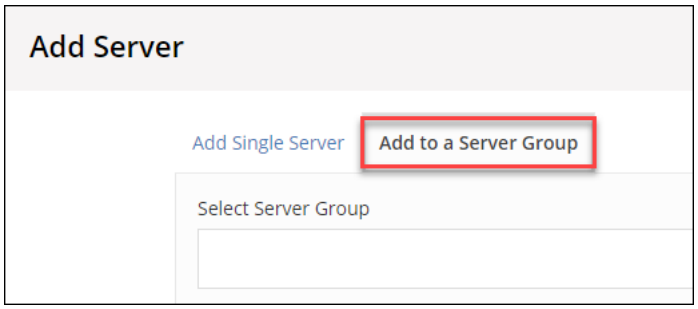
Enter then the server information. Click **Add** when your entries are complete.

If you want to add more servers at this time, click **+ Add** and follow the steps.

When your new server group is complete, click **Save**.

Adding a Corticon Server to an existing group

When you choose **Add a single Corticon Server**, the **Add Server** dialog box provides an **Add to a Server Group** tab so you can add the server to an existing group:



Click **Select Server Group** to choose a group, and then enter the server information. When your new server and its group assignment are complete, click **Save**.

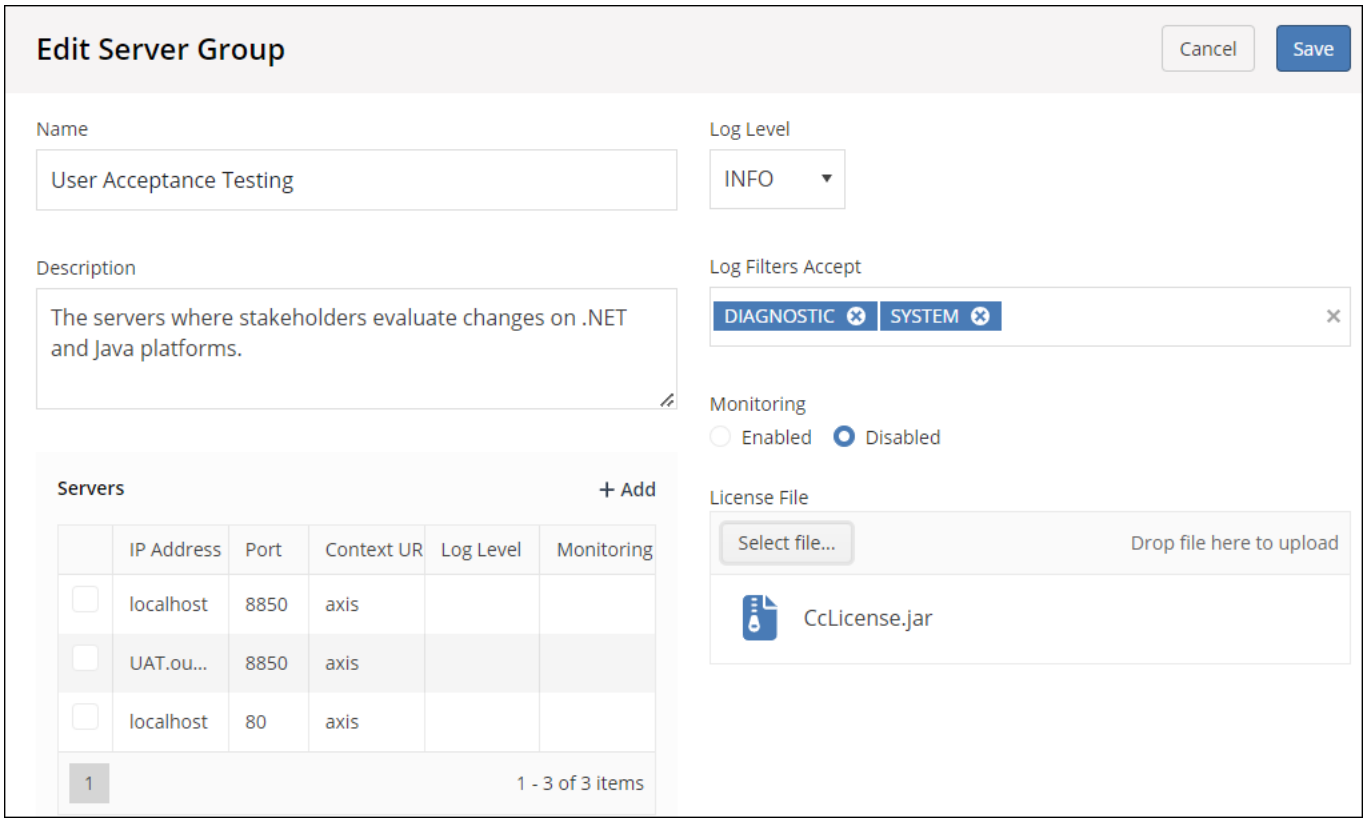
Note: Adding a server -- individually or within a group -- as `localhost` might seem practical during evaluation and testing, but when you access Web Console from a remote machine that has a server installation that you want to add, you might find that references to `localhost` are distracting as it is not *this* localhost. It is a good practice to always use DNS-resolvable hostnames or static IP addresses.

Edit Server groups and Servers

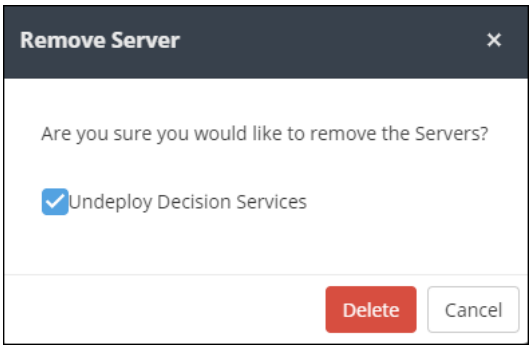
After adding a Server group or a server, you can change its configuration.

To edit a Server Group:

Select **Edit** on the server group's **Details** page to open its edit dialog box:



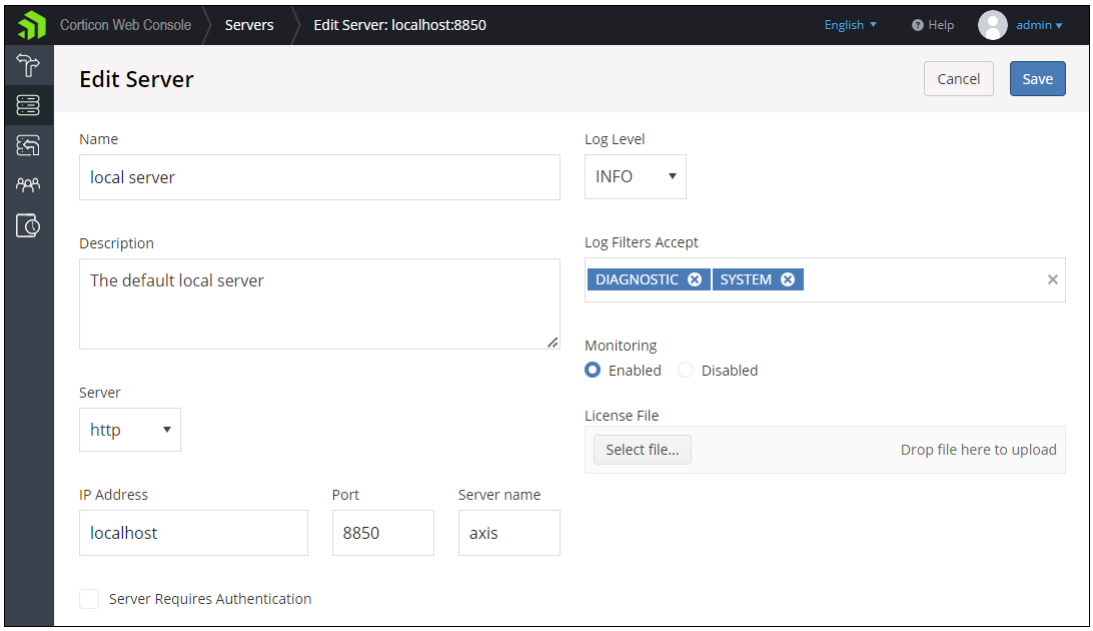
1. Edit the name and description as appropriate
2. Click **+ Add** to add more servers.
3. Select a server to access its edit and delete functions. **Edit** lets you change the server information. **Delete** removes the selected servers from the group and the Web Console. You are asked to decide whether to undeploy any Decision Services before deletion, and then confirm the deletion action:



   Note that deleting servers from the Web Console does not stop or delete the actual running server instance; it just removes the registration of the server with the Web Console. The server continues to run and could be added back to the Web Console.
4. You can change other server properties that will apply to all servers in the group as illustrated on the right side of the dialog box: **Log Level**, **Log Filters Accept**, **Monitoring**, and **License File**.

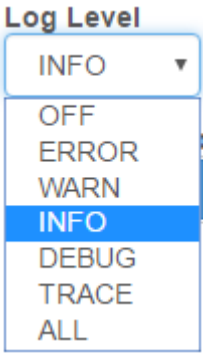
To edit a Server:

Select **Edit** on the server's **Details** page to open its edit dialog box:



Edit:

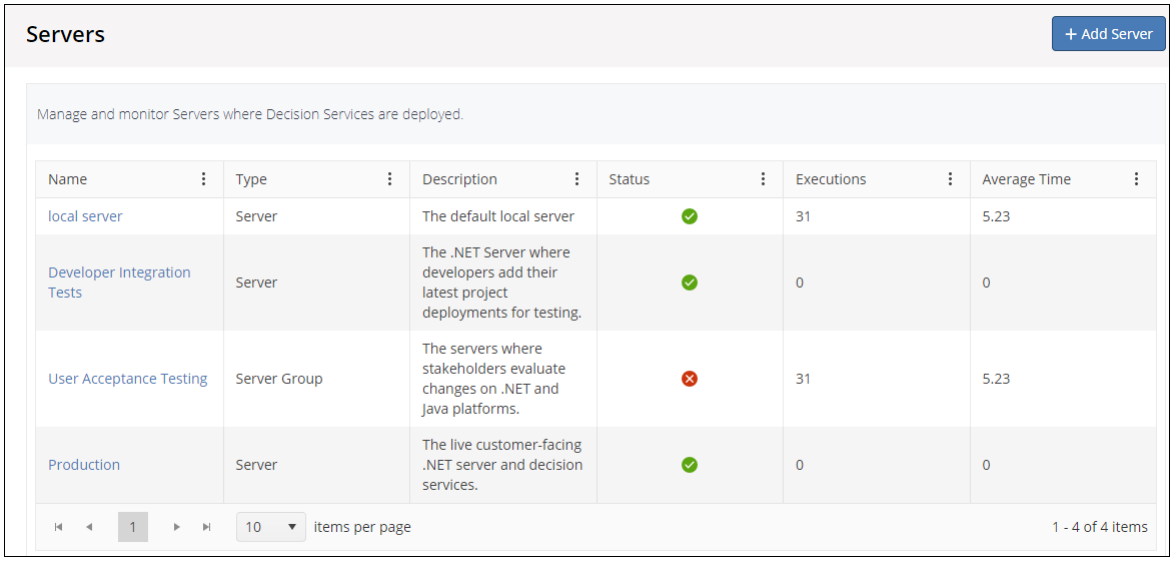
- **Name**
- **Description**
- **Server** hostname/IP address, port, and context URL
- **Log level** - The log level on the selected server. The default level is INFO. When you change the level and save the edits, it is immediately applied to that server without stopping and restarting the server. The logs promptly reflect the changed level of detail.



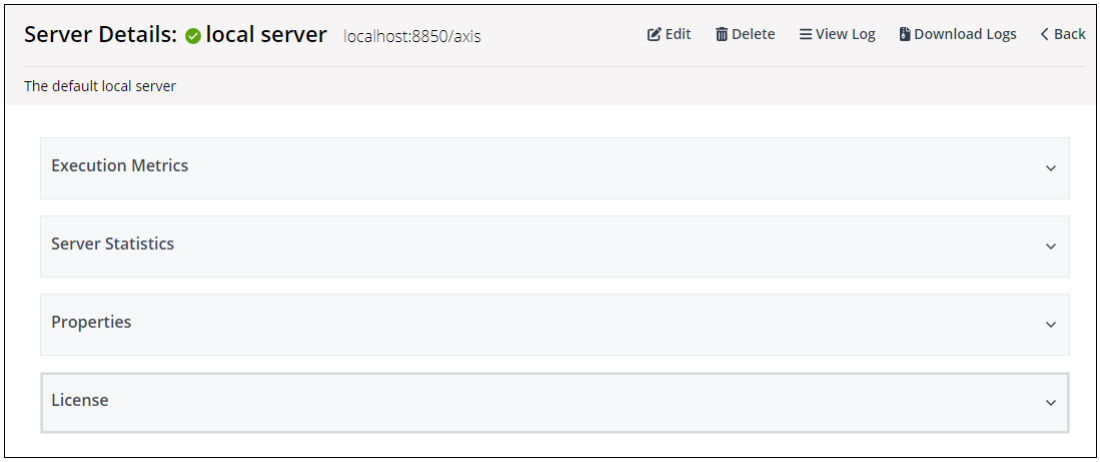
- **Monitoring** - Determines whether the statistics from this server are gathered by the Web Console and stored for later analysis.
- **License File** - Copies the selected `CcLicense.jar` (or its preferred name) from the machine where the browser is connected to the Web Console (or a network-accessible location) to the `CcServerSandbox` on the machine hosting this server.

Explore Server features

When you click the **Servers** button:  in the left panel, the servers and server groups are listed. In this example, there is one Server Group and three Servers:



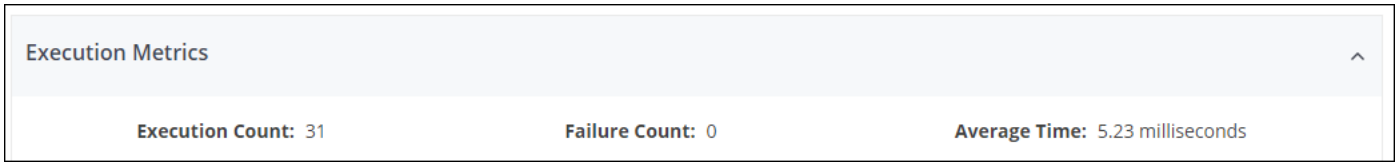
Clicking on a server or server group **Name** selects it, and then opens its **Details** page to display the deployment and operational information about it.



For the selected Server, you can choose **Edit**, **Delete**, **View Log**, or **Download Logs**.

Server Execution Metrics

Execution metrics provide counts and performance data of all Decision Services running on the selected server, or aggregated across a server group.

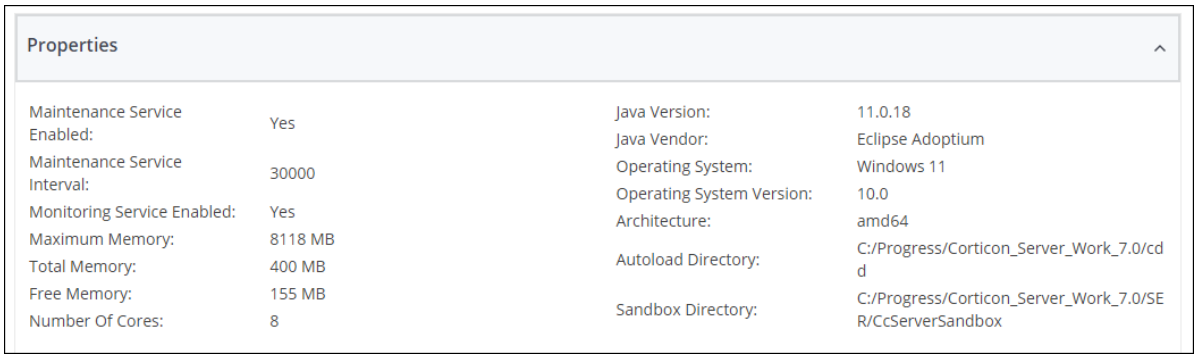


Server Statistics

You can look at metrics and statistics at several levels from for all Decision Services running on server or aggregated for all Decision Services and Servers in a server group. The following view shows the categories of information for a server group:

Properties

A Server's **Properties** lists important settings and platform environment data of the server, from its point-of-view:



Properties are specific to a Corticon Server on the machine where it is installed and running. They are accessed for an individual server, or a member of a server group.

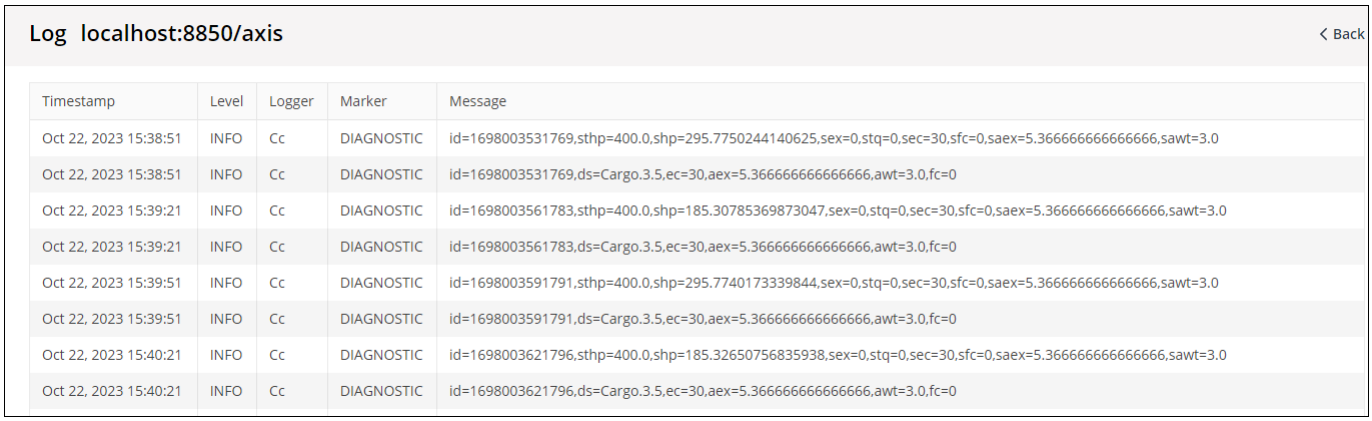
License

License information shows the location of the Corticon license that a specific server is using, as well as essential information about that license:

The license file that enabled the server to run is typically updated only when a new license has been provided that changes the expiration and enabled features for that server.

View log

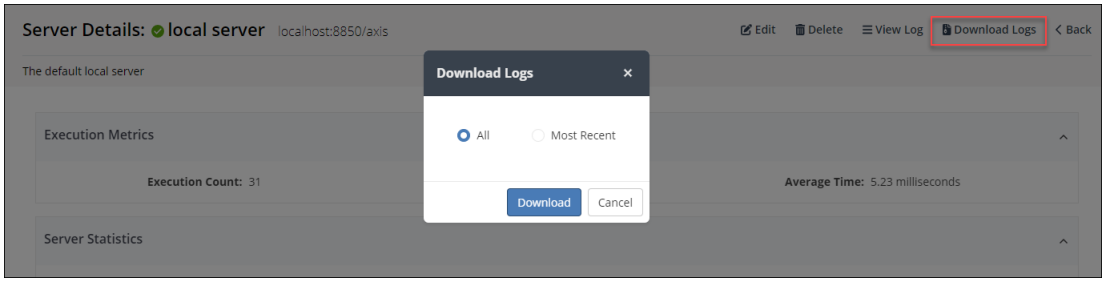
Servers lets you access the tail of the current `CcServer.log` file that the server is using:



Timestamp	Level	Logger	Marker	Message
Oct 22, 2023 15:38:51	INFO	Cc	DIAGNOSTIC	id=1698003531769,sthp=400.0,shp=295.7750244140625,sex=0,stq=0,sec=30,sfc=0,saex=5.366666666666666,sawt=3.0
Oct 22, 2023 15:38:51	INFO	Cc	DIAGNOSTIC	id=1698003531769,ds=Cargo.3.5,ec=30,aex=5.366666666666666,awt=3.0,fc=0
Oct 22, 2023 15:39:21	INFO	Cc	DIAGNOSTIC	id=1698003561783,sthp=400.0,shp=185.30785369873047,sex=0,stq=0,sec=30,sfc=0,saex=5.366666666666666,sawt=3.0
Oct 22, 2023 15:39:21	INFO	Cc	DIAGNOSTIC	id=1698003561783,ds=Cargo.3.5,ec=30,aex=5.366666666666666,awt=3.0,fc=0
Oct 22, 2023 15:39:51	INFO	Cc	DIAGNOSTIC	id=1698003591791,sthp=400.0,shp=295.7740173339844,sex=0,stq=0,sec=30,sfc=0,saex=5.366666666666666,sawt=3.0
Oct 22, 2023 15:39:51	INFO	Cc	DIAGNOSTIC	id=1698003591791,ds=Cargo.3.5,ec=30,aex=5.366666666666666,awt=3.0,fc=0
Oct 22, 2023 15:40:21	INFO	Cc	DIAGNOSTIC	id=1698003621796,sthp=400.0,shp=185.32650756835938,sex=0,stq=0,sec=30,sfc=0,saex=5.366666666666666,sawt=3.0
Oct 22, 2023 15:40:21	INFO	Cc	DIAGNOSTIC	id=1698003621796,ds=Cargo.3.5,ec=30,aex=5.366666666666666,awt=3.0,fc=0

Download log

Corticon Web Console enables you to download and view Server log files. This is especially useful when you need to locally examine a remote Server's log files to identify the source of a problem. To download a Server's log files, click **Download Logs** in the Server page. This opens a dialog box where you can choose to download **All** log files or only the **Most Recent**. If you choose **All**, all log files that have been retained since installation will be downloaded. If you choose **Most Recent** , you will get all log files that have been modified by the Server in the last 24 hours. Select the appropriate option and click **Download**. This downloads a ZIP file named `CcServerLog.zip` that contains the Server log files.



To know more about troubleshooting a problem by examining Server logs, see *Troubleshooting Corticon Server in the Server Guide*.

Decision Services and Applications

Types of Decision Services

Many Decision Services might be deployed on a Corticon Server. There are two types of Decision Services from the point of view of the Web Console, based on how they were deployed:

- **Managed Decision Services** are those deployed through the Web Console. For managed Decision Services the Web Console has the EDS file, and can perform more management activities such as deploying it to additional Corticon Servers. Managed Decision Services can be:
  - Added directly through the Web Console's **Add Decision Service** feature.
  - Added directly from Corticon Studio using the Studio's **Package and Deploy** feature. Studio prompts for the Application where the Decision Service will be added, and the Server or Server Group where it will be deployed.
- **Discovered Decision Services** are those deployed not through the Web Console but through another means. The management operations the Web Console can perform on discovered Decision Services is limited so as not to conflict with how they were deployed. Discovered Decision Services could be:
  - Decision Services packaged and deployed directly from Corticon Studio or any of the deployment tools.
  - These are Decision Services deployed through Corticon Deployment Descriptors (CDDs) -- text-based files that specify a Decision Service to be deployed and its deployment properties. CDD files are automatically loaded by the Corticon Server.

In most deployments, you will likely use either managed or unmanaged Decision Services. The approach you take for deployment and management depends on your needs.

Opening the Decision Services and Applications page

1. Connect to the Web Console server where you manage Decision Services.
2. Click the **Decision Services** button: 

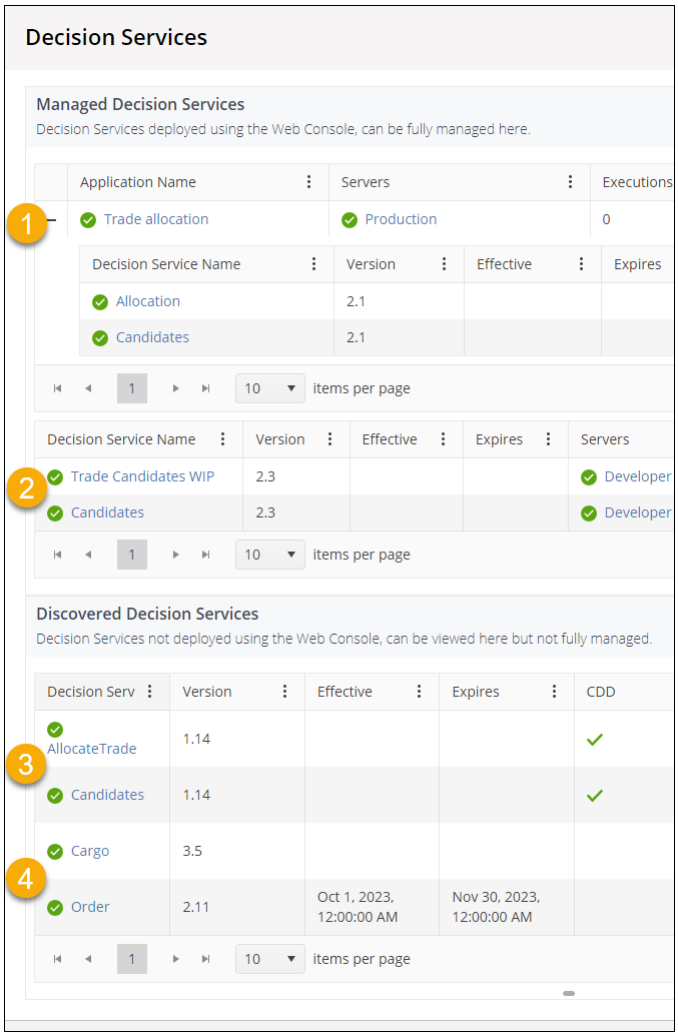
The Decision Services page shows all the types of Decision Services on the managed servers.

How Decision Service types are displayed

Decision services are identified as either managed (created in the Web Console) or discovered. Managed Decision Service show applications with their Decision Services. Discovered Decision Services identify the ones that were using CDDs to identify their source.

In the illustration:

1. Managed Applications
2. Managed Decision Services
3. Discovered Decision services from CDDs
4. Discovered Decision Services



How to use Applications

An Application is a group of Decision Services that you can deploy to a Server or Server Group. When you deploy an Application to a Server Group, all Decision Services in the Application are deployed to each of the Corticon Servers in the Server Group. Further, if a new server is added to the Server Group, the Web Console automatically deploys the Application to it. An Application is therefore, a unit of deployment. It enables you to manage a set of related Decision Services more easily.

In order to add a Decision Service to an Application, you need to have a Decision Service file (`.eds`) that was packaged from a Ruleflow. There are several toolsets that perform this task, as discussed in the section *"How to package and deploy Decision Services" in the Deployment Guide*.

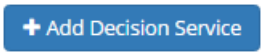
A feature of Corticon Studio, as described in *"Deploy to Corticon Web Console" in the Deployment Guide*, enables you to select Ruleflows in a project to deploy as Decision Services that are sent to a new or existing Application assigned to a server or server group managed in a Web Console. As a result, the Decision Services are immediately deployed (or redeployed) to the server or all active servers in the Server Group.

Add or Edit a Decision Service

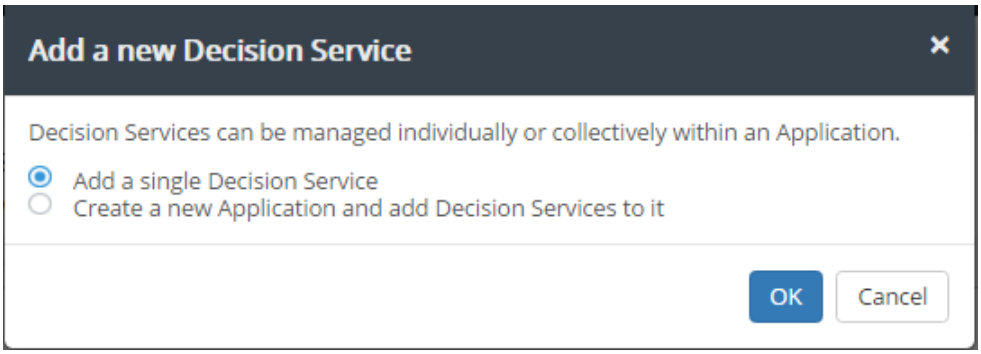
The following procedures show to bring a Decision Service under management either as an independent Decision Service, or as a member of an Application.

Note: As the general steps are common to both adding and editing a Decision Service, this topic focuses on the tasks when adding a Decision Service, and then shows how to access a Decision Service to edit it.

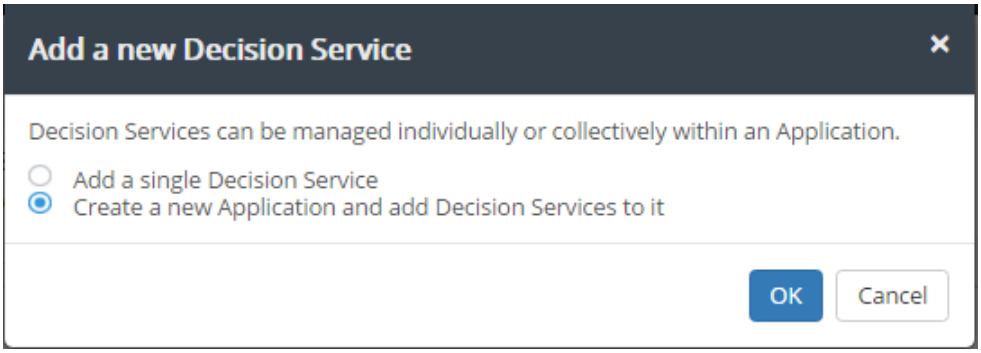
To add a Decision Service:

1. Connect to the Web Console server where you want to add Decision Services.
2. Click the **Decision Services** button: 
3. Click **+ Add Decision Service**: 

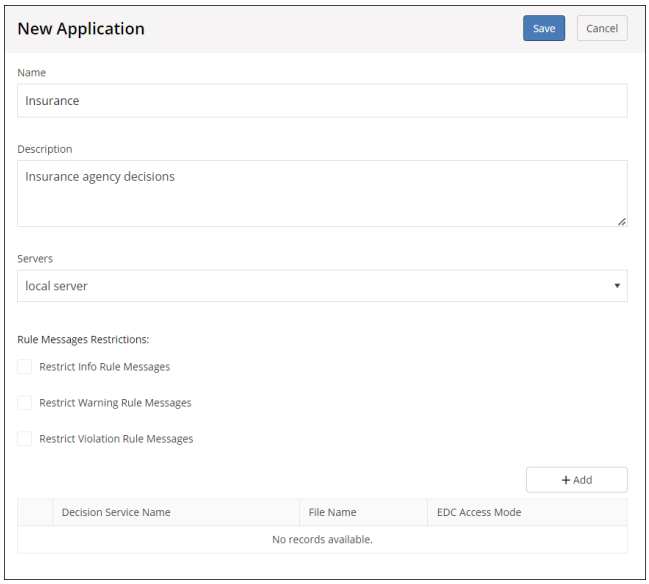
4. The **Add a new Decision Service** dialog box opens:



5. You can choose to create an Application for the Decision Service you are adding:



   a. If you choose that option and click **OK**, the **New Application** dialog opens:



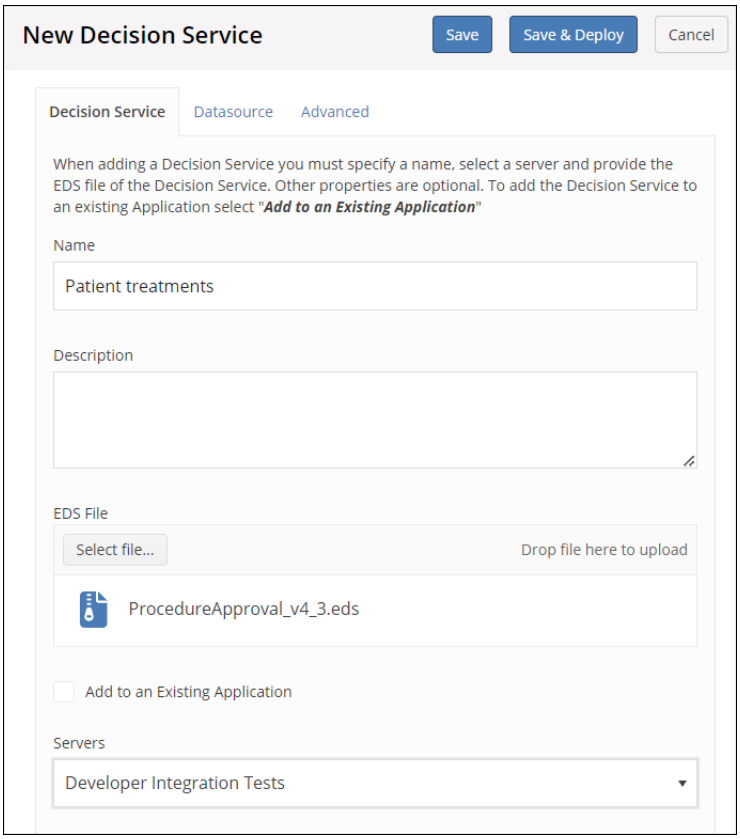
b. Enter a Name and Description.

c. Choose the server or server group where the Application's Decision Services are to be deployed.

d. Set options that will apply to all Decision Services in the Application.

e. Click **+ Add** for each Decision Service you want to add to the Application.

6. On either path, the **New Decision Service** dialog box opens at the **Decision Service** tab:



a. Enter a name. Note that this will be its name when deployed, not the name of the EDS file you choose.

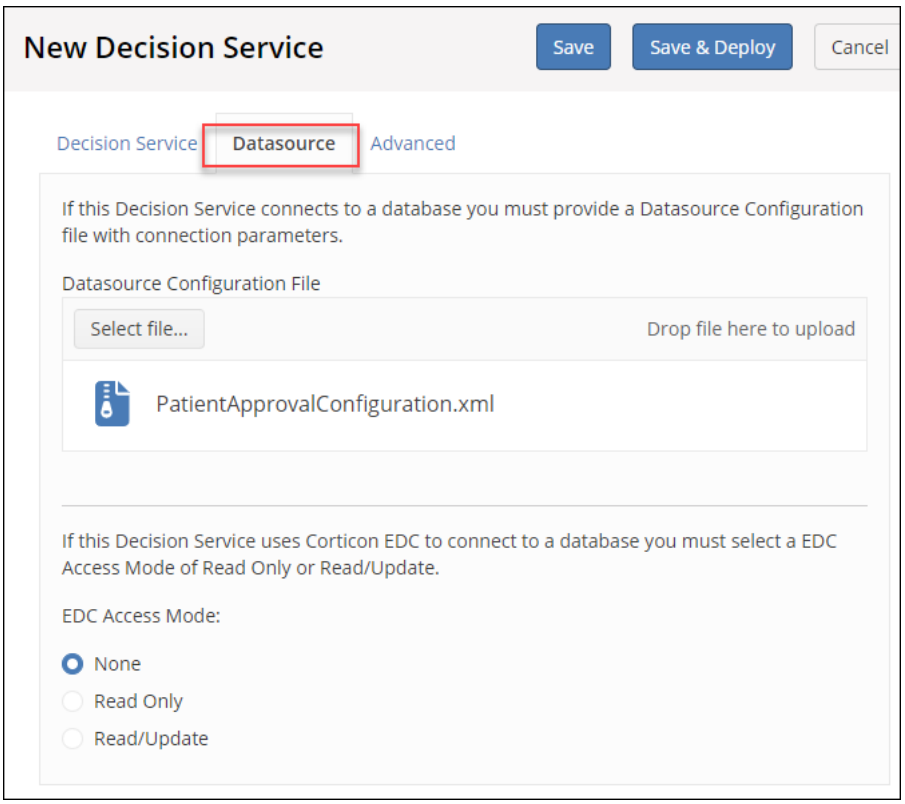
b. Add a description.

c. Click **Choose file** to locate an EDS file.

d. Choose a server or server group

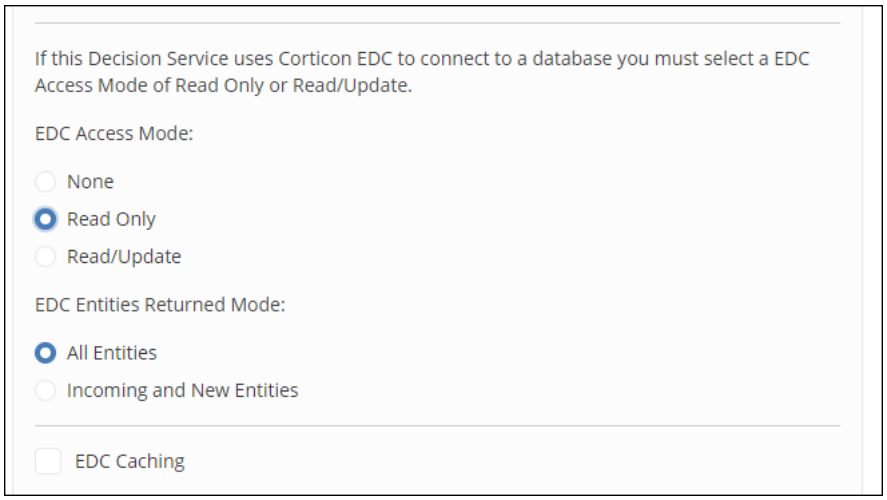
e. If you started this process as a single Decision Service, you can choose to add it to an existing application from the list that will be offered.

7. Click the **Datasource** tab to access its options:



a. **Datasource Configuration File**: Specify the XML file that contains the data source access properties. To learn how to generate this file from Corticon Studio, see the topic: *"Export the Datasource Configuration file" in the Data Integration Guide*.

b. You can change the **EDC Access Mode** option to either **Read Only** or **Read/Update** to extend the dialog tab to display additional configuration settings:



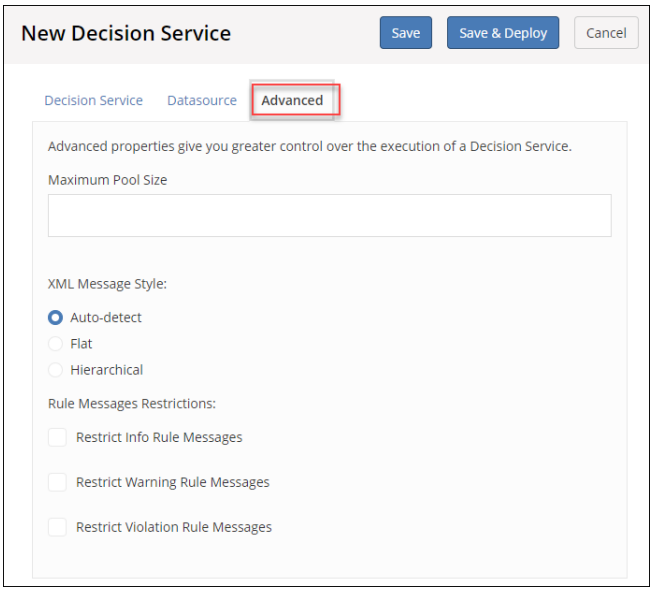
c. In the **EDC Access Mode**, choose the appropriate access option. This setting controls how a Decision Service will access connected databases. Select **Read Only** or **Read/Update** to then expose additional settings that you need to configure:

- **EDC Entities Returned Mode**: Choosing **All Entities** returns all records from the database when the Decision Service executes. Choosing **Incoming and New Entities** returns entities that were in the request message and only those entity records that are added or modified in the database when the Decision Service executes. Select the appropriate option.
- **EDC Caching**: Database caching enables Corticon to store often-used data in a cache. This improves the performance of the Decision Service since it can read and write data in the cache faster than if this data was in the database. If you choose **Enabled**, database caching will be enabled for the Decision Service. To learn more about database caching, see the topic: *"How to work with database caches" in the Data Integration Guide*.

Important: Turning caching on or off - If you want to enable or disable caching on a deployed Decision Service, the mechanisms of caching require that you undeploy and delete the Decision Service, and then add and deploy the Decision Service again with the cache enablement setting you want.

---

8. Click the **Advanced** tab to access its options:



Consult the *Server Guide* for more information about these settings.

a. In the **Maximum Pool Size** field, specify how many execution threads for this Decision Service will be added to the execution queue. If you leave this field blank, the Web Console will set a default value of **1**.

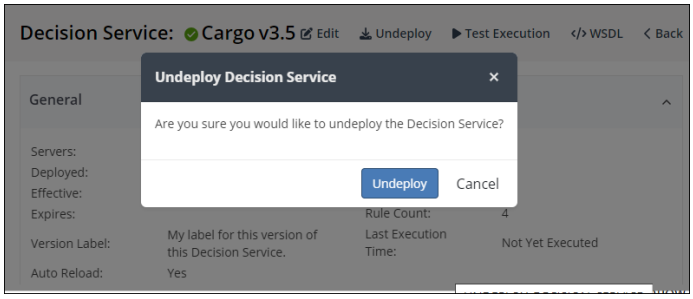
b. In the **XML message style** section, choose whether request messages for this Decision Service should contain a **Flat** or **Hierarchical** payload structure. **Auto Detect** accepts either style.

9. Click **Save** to store the Decision Service but not deploy it. Click **Save & Deploy** to store the Decision Service and also deploy it. Click **Cancel** to close without making changes.

Undeploy a Decision Service on a Server

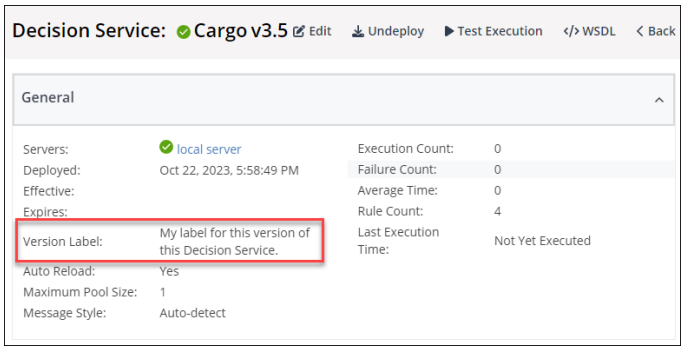
You can undeploy Decision Services by selecting the Decision Service and clicking **Undeploy**. Performing this operation on a managed Decision Service takes you to the Application details page, which has options to remove individual Decision Services or undeploy the Application altogether.

Note: In the Web Console, you cannot undeploy a Decision Service that was deployed using a CDD file.

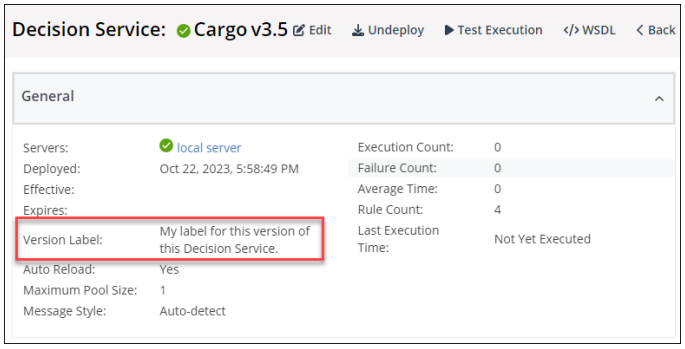


Decision Service General Information

The **Version** Label is the text that was added to the Ruleflow Properties that generated the Decision Service.



Effective and **Expires** were added as the **Effective Date** and **Expiration Date** on the Ruleflow that generated the Decision Service.



General metrics are a simple table of the count of all request executions of a Decision Service on the selected server, the count of failures, and the average execution time. The average time is average execution time for execution of all the Decision Services on this server.

Note: The metrics are reset when a server restarts.

Simplified JSON in requests

Some users find that their JSON requests have metadata only at the root, expecting that the decision service can infer the metadata for subordinate levels. That tactic is supported, although the output provides the metadata at all levels. The following example shows the `coupons` Ruletest in the advanced tutorial. It executes as expected with either request, and produces identical responses (both with metadata).

How to: Use Simplified JSON in requests

1. Copy the request you want to simplify to an editor.
2. Add the decision service `name` as name at the top.
3. Delete (or leave out) the metadata for related entities.
4. Retain the metadata for the primary entity.

For example:

coupons.json

```
{
    "__metadataRoot": {"#locale": ""},
    "Objects": [{
        "preferredCard": {
            "cumulativeCashBack": 9.24,
            "__metadata": {
                "#type": "PreferredAccount",
                "#id": "PreferredAccount_id_1"
            },
            "cardNumber": "12"
        },
        "ShoppingCart": {
            "totalAmount": null,
            "Item": [
                {
                    "price": 25,
                    "name": "Miller Beer",
                    "__metadata": {
                        "#type": "Item",
                        "#id": "Item_id_1"
                    },
                    "department": null,
                    "barCode": "39-291-1234"
                },
                {
                    "price": 1.99,
                    "name": "RedBull",
                    "__metadata": {
                        "#type": "Item",
                        "#id": "Item_id_2"
                    },
                    "department": null,
                    "barCode": "39-285-98765"
                }
            ],
            "useCashBack": true,
            "savings": null,
            "__metadata": {
                "#type": "ShoppingCart",
                "#id": "ShoppingCart_id_1"
            }
        },
        "__metadata": {
            "#type": "Customer",
            "#id": "Customer_id_1"
        },
        "Name": ""
    }]
}
```

coupons_Simplified.json

```
{
    "__metadataRoot": {"#locale": ""},
    "name" : "MyAdvancedTutorial",
    "Objects": [{
        "preferredCard": {
            "cumulativeCashBack": 9.24,
            "cardNumber": "12"
        },
        "ShoppingCart": {
            "totalAmount": null,
            "Item": [
                {
                    "price": 25,
                    "name": "Miller Beer",
                    "department": null,
                    "barCode": "39-291-1234"
                },
                {
                    "price": 1.99,
                    "name": "RedBull",
                    "department": null,
                    "barCode": "39-285-98765"
                }
            ],
            "useCashBack": true,
            "savings": null
        },
        "__metadata": {
            "#type": "Customer",
            "#id": "Customer_id_1"
        },
        "Name": ""
    }]
}
```

Note

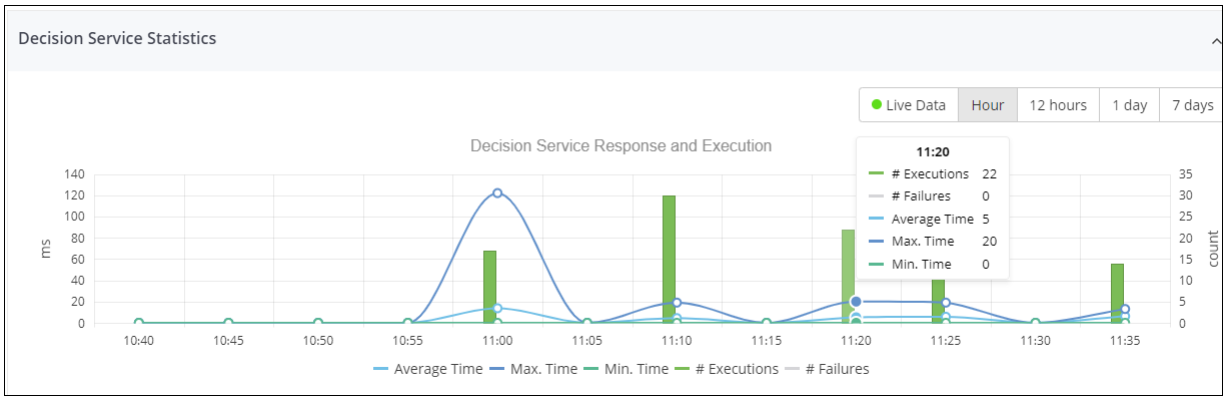
- In the example, `Name` at the bottom is an attribute of the root entity.
- Mind your commas, braces, and brackets when you are paring down a known-good request.

Decision Service Details

Click on a Decision Service to display its operational and performance data.

If the Decision Service is deployed to a Server Group, the operational and performance data is an aggregate of that Decision Service from all servers in the server group.

The actions available let you **Edit**, **Delete**, **Undeploy**, **Test Execution**, and display **WSDL**.



You can collapse and expand sections of the page to manage the display.

Application Details

The general metrics shown for an application are a rollup of the metrics of the Decision Services in the Application. For example, the average execution time shown on an application is the average execution time of all it Decision Services.

Test Execution

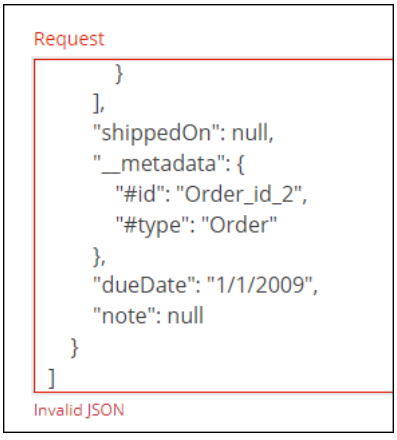
The **Test Execution** option lets you test your Decision Service by making a REST or SOAP request to it. When you select the Test Execution, you choose a server where the Decision Service is deployed, whether to make a REST or SOAP request, and then locate a JSON or XML file for the payload of the request.

Note: While the Decision Service name is essential for Corticon requests, this panel ignores the `decisionServiceName` parameter in the request as it is focused on the current Decision Service.

To execute a test against a selected deployment of the current Decision Service :

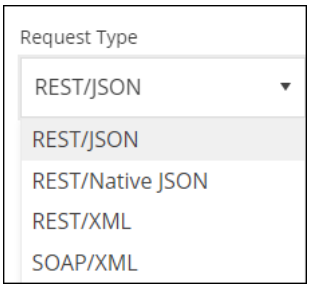
1. Click **Server** to select a server that has the Decision Service deployed.
2. In the **Choose Request File** area, click **Choose File**, then locate and open an XML or JSON request appropriate for the Decision Service. The **Request** area shows the request text.

Note: You could choose to paste or type in the request. The entry will be evaluated as to whether it is valid, and, if invalid, highlights the Request area, as illustrated:



---

3. Choose its **Request Type** will attempt to set the type from the selected file:



where:

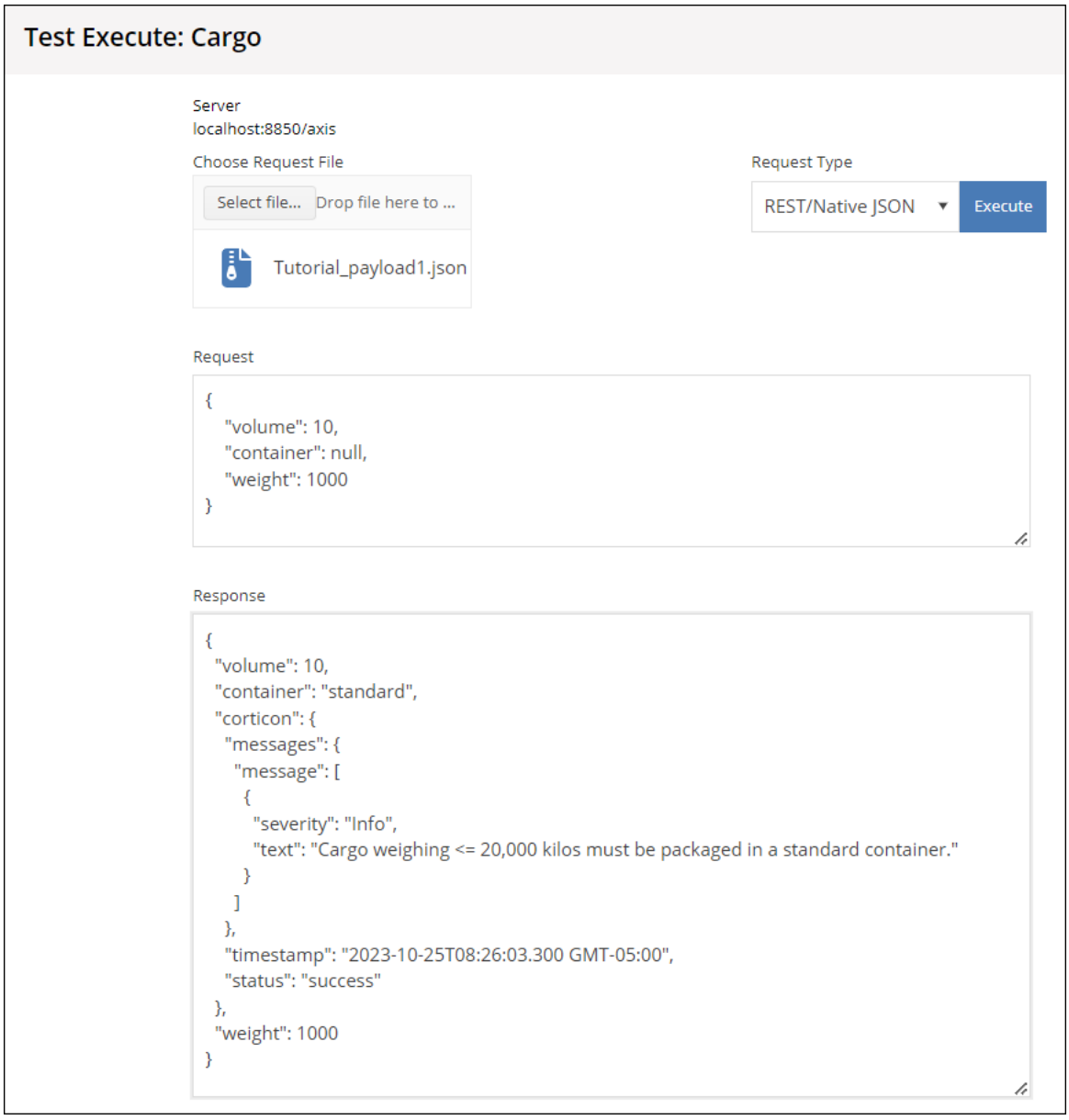
- **REST/JSON** expects well-formed JSON.
- **REST/NativeJSON** expects the request to adhere to simplified format of Native JSON. It does not handle arrays, so only one top-level entity will be valid.
- **REST/XML** expects well-formed XML.
- **SOAP/XML** expects well-formed XML in a SOAP wrapper.

Note: XML requests require the decision service name to be the name of the Decision Service in the Web Console.

---

4. Click **Execute**.

The request executes, and then adds the **Response** text, as shown:

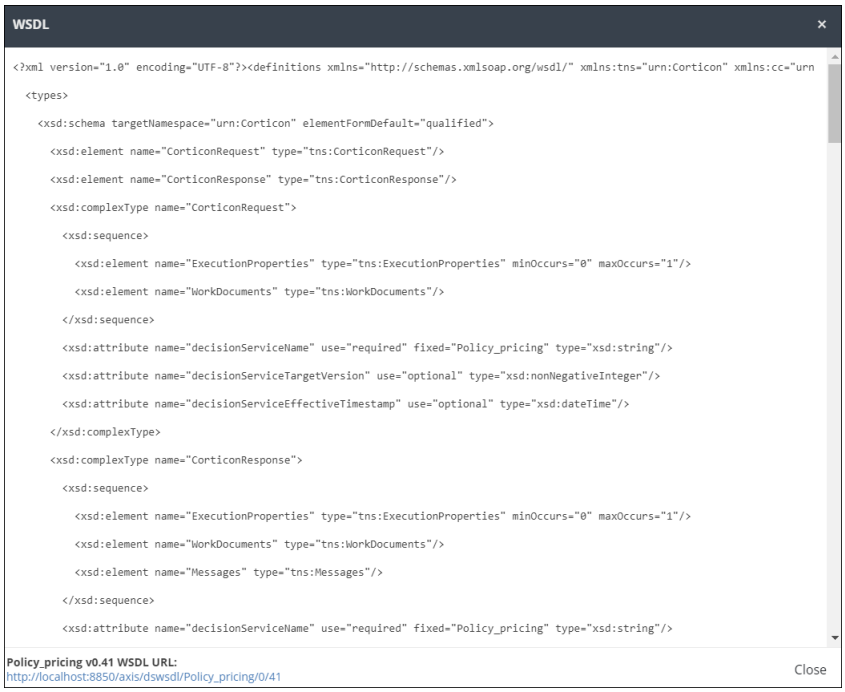


WSDL

The **WSDL** option displays the current Decision Service's WSDL, and also provides a link to WSDL data in an editor.

Note: Decision Service compilation does not, by default, include WSDL and its report in the EDS file by default. If you use WSDL or reports in the EDS, add the following lines to the `brms.properties` file where Corticon will do the compilation:

```
com.corticon.server.compile.add.wsdl=true
com.corticon.server.compile.add.report=true.
```



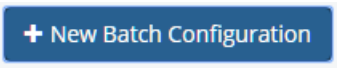
Batch Configurations

Corticon Web Console lets you connect to remote Web Console servers that in turn connect to managed Corticon Servers where deployed Decision Services are defined that integrate with data sources. When these Decision Services use defined SQL batch queries linked through the Datasource, you can define batch configurations and run batch jobs.

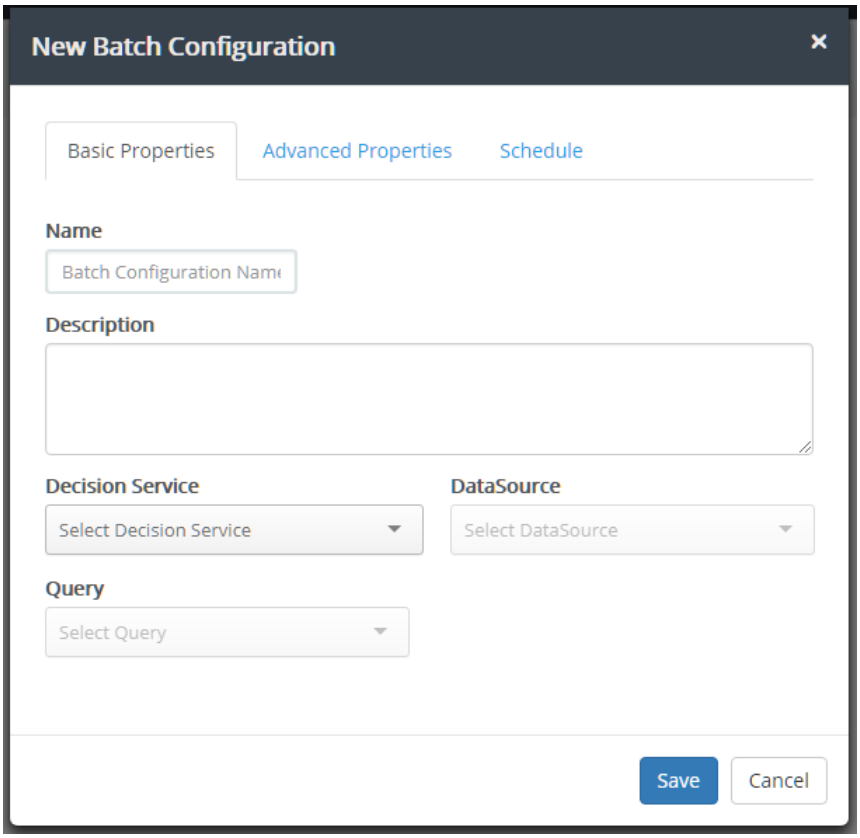
As a result, you can ensure that high-volume rules-based processing occurs on a specified schedule.

Add Batch Configurations

To add batch configurations:

1. Connect to the Web Console server where you maintain batch configurations.
2. Click the **Batch configurations** button: 
3. Click **+ New Batch Configuration**: 

4. The **New Batch Configuration** dialog box opens:



where:

- **Name** - Unique text that you want to use to refer to this configuration
- **Description** - Optional supporting text for the configuration
- **Decision Service** - List of managed, deployed Decision Services that have at least one component that has batch queries in its connected database.
- **Datasource** - The name of the Datasource connection that the Decision Service uses, as assigned in the Vocabulary.

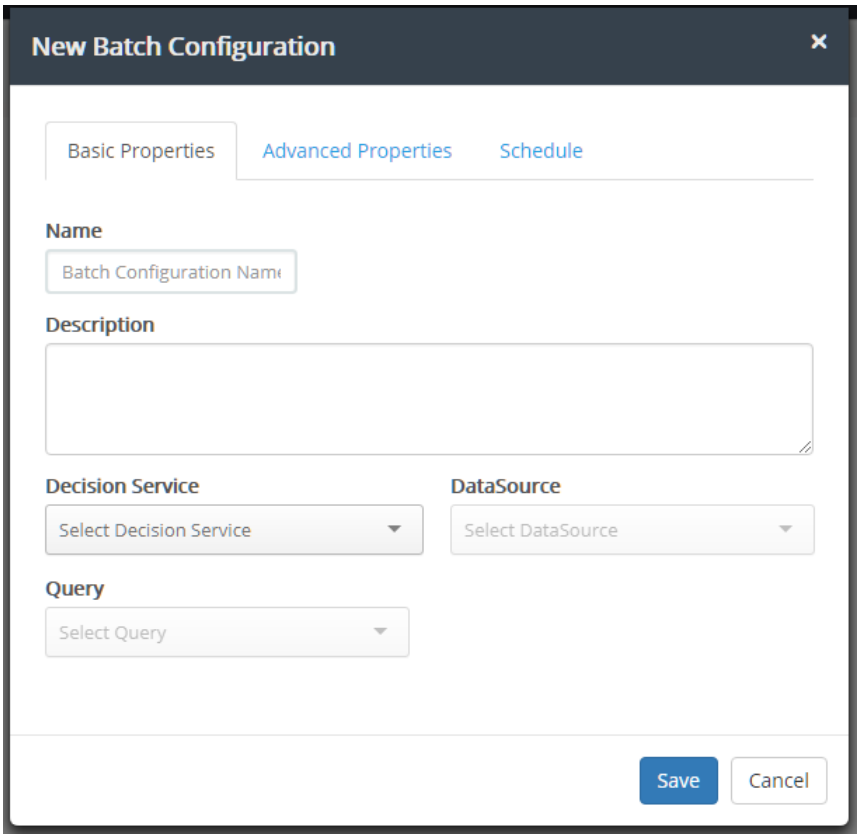
  For example, in an export configuration file named `myConfig.xml` where the first few lines are...

  ```
  ...
  <decisionService>
     <datasources>
        <database useForQueryService="true" name="Patient Data">
           <connection-url>jdbc:progress:sqlserver://localhost:1433;
               databaseName=PatientRecords</connection-url>
  ```

  ...the **Datasource** value is `Patient Data`.

- **Query** - The name of the batch query stored in one of the Decision Service's connected databases

5. Click to access the **Advanced Properties** tab:



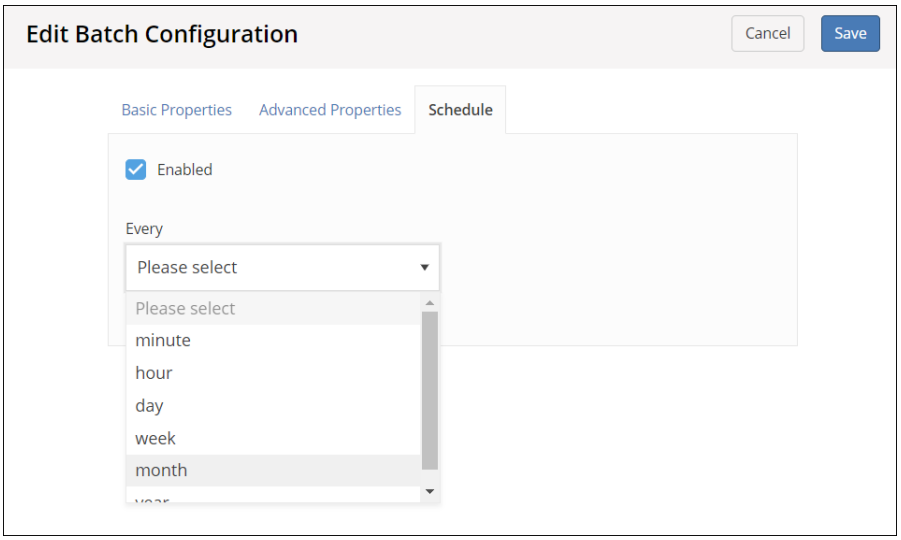
where:

- **Number of ID's per Fetch** - Number of Ids that will be retrieved by each Datasource Fetch. Default value is `1000`.
- **Entities per Payload** - Number of entities that will be added to each payload sent to the Corticon Server execute method. Default value is `1`.
- **Number of Processing Threads** - The number of execution threads the Corticon Server will spawn when executing the batch. The Default value is the number of cores on the Corticon Server's machine.
- **Log Path** - The folder that will store the logs produced for this batch configuration on the server that runs the batch process. Default location is `[CORTICON_WORK_DIR]\logs\` .

  The log file name is set as *`DecisionServiceName(Version)_Threads_Timestamp`*`.log`. For example, `PatientUpdate(1.2)_4_1515014748084.log`

- ***Logging enabled checkbox*** - To the right of the **Log Path** entry, the checkbox lets you decide whether to do logging for this batch configuration.

6. Click to access the **Schedule** tab:



where:

- **Enabled** - Chooses to repeat the batch process with the frequency you specify.
- *Choose Frequency:*
  - **minute** - Once every minute.
  - **hour** - At specified minute past every hour.
  - **day** - At the specified time of every day.
  - **week** - At specified week day at the specified time of that day.
  - **month** - At specified day every month at the specified time of that day.
  - **year** - At specified day and month every year at the specified time of that day.

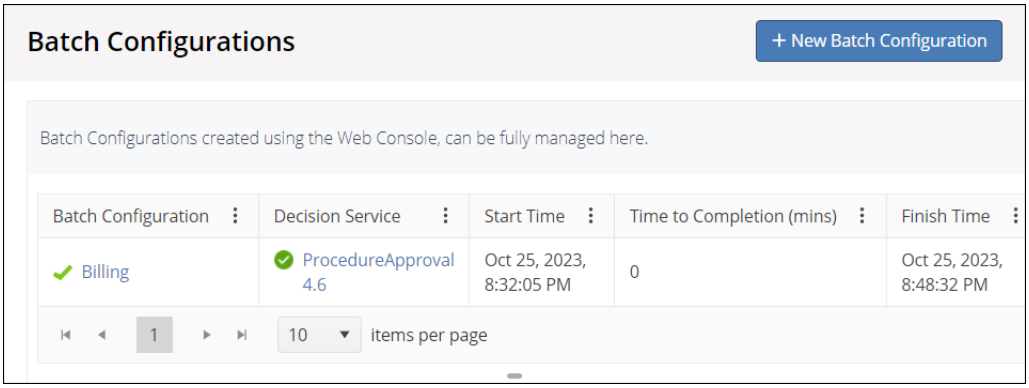
Note: On most of the schedule options, you can use CTRL+click to choose multiple values.

7. Click **Save**.

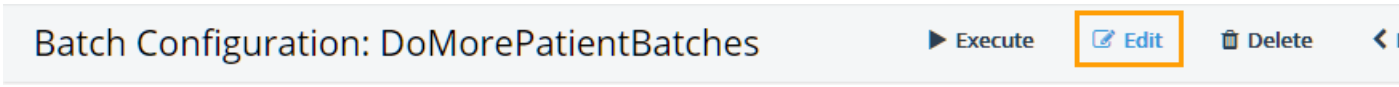
Edit Batch Configurations

To maintain batch configurations:

1. Connect to the Web Console server where you maintain batch configurations.
2. Click the **Batch configurations** button:  The Batch Configuration page opens and displays the current batch configurations, as illustrated:



3. Click the Batch Configuration name you want to edit.
4. On the Details page, click **Edit**, as shown:



5. The **Edit Batch Configuration** dialog box opens.
6. Follow the steps for the dialog box as described in Add Batch Configurations on page 36
7. Click **Save**.

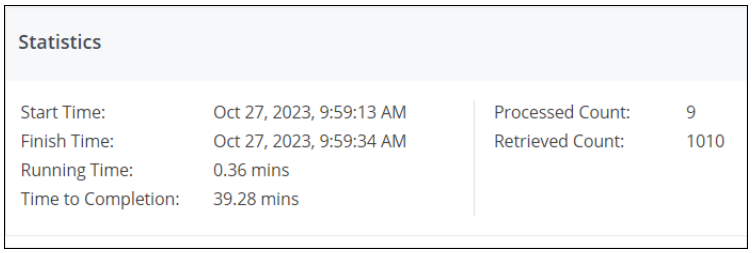
Run Batch Configurations

To run a batch configuration:

1. Connect to the Web Console server where you maintain batch configurations.
2. Click the **Batch Configurations** button:  The Batch Configuration page opens and displays the current batch configurations.
3. Click the Batch Configuration name you want to edit.
4. On the Details page, click **Execute**.

Note: When an execution is running, you can terminate it by clicking **Stop**

The job statistics show the time and counts of the most current run, as shown:



The logs are produced on the server that ran the deployed Decision Service at the location you specified or the default location `[CORTICON_WORK_DIR]\logs`. The filename for each run is *`DecisionServiceName(Version)_Threads_Timestamp`*`.log`

Note: For an example of batch runs, see *"Getting Started with Batch" in the Data Integration Guide.*

For more about batch processing in Corticon, see:

- *"Getting Started with Batch" in the Data Integration Guide.*

How to view the Activity Log

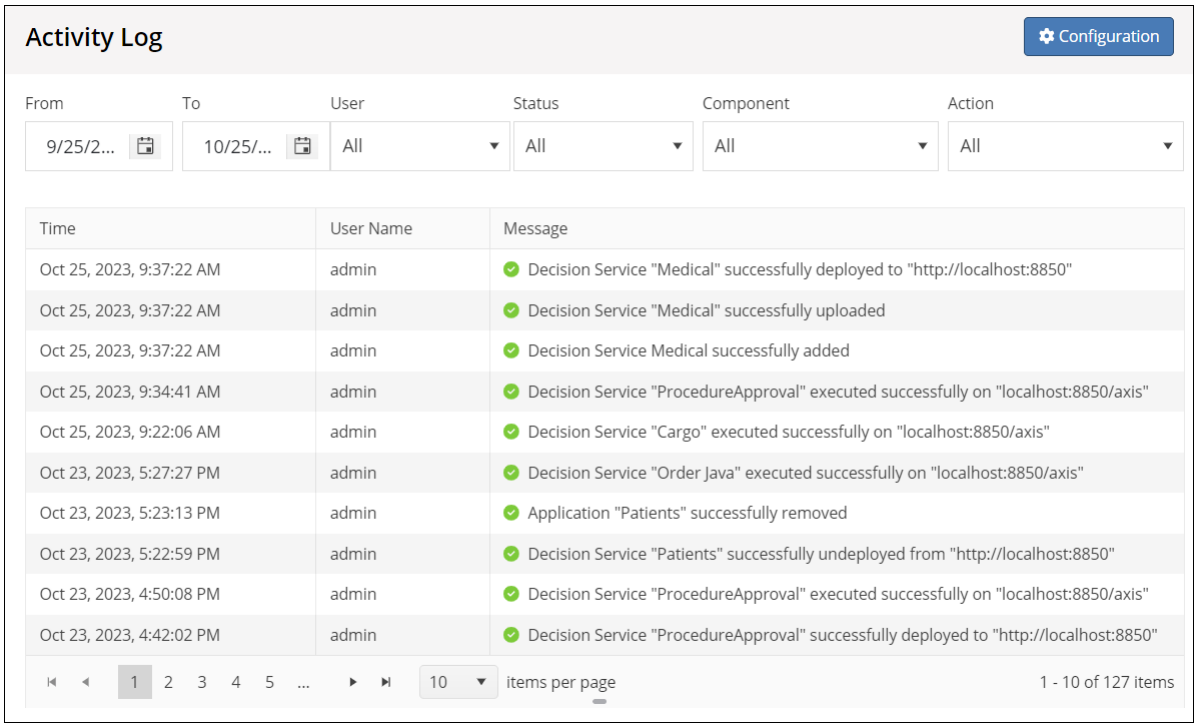
Corticon Web Console maintains a log of its activities. The log includes:

- User actions such as deploying or undeploying Decision Services and creating or modifying Applications and Servers.
- System events such as deployment failures and lost connections to Servers.

To view the activity log:

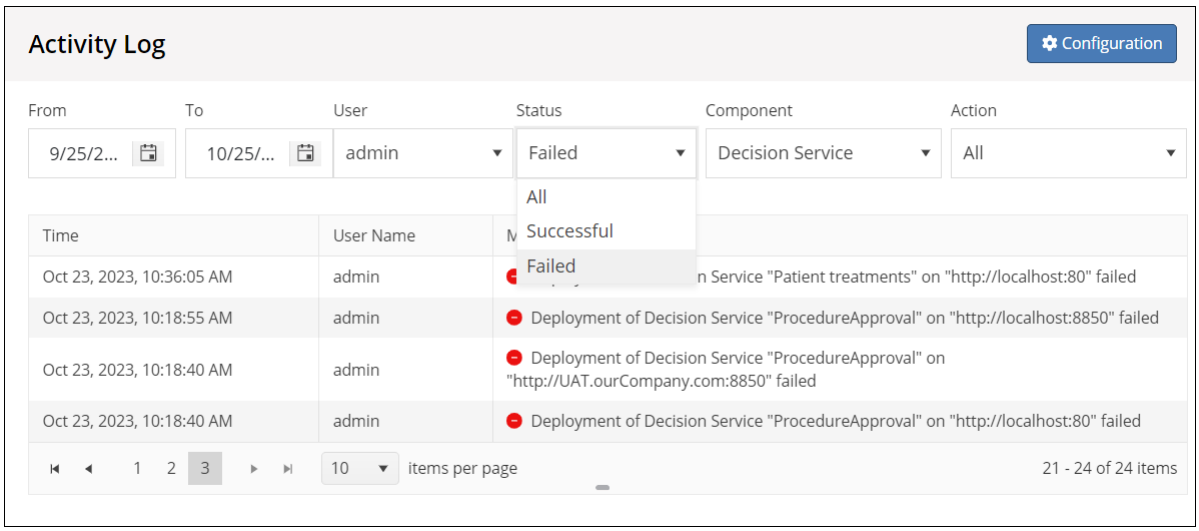
1. Connect to the Web Console server where you want to view the Activity Log.
2. Click the **Activity Log** button: 

The Activity Log page opens and displays the log in a three-column table:



Some log messages, such as those relating to failed deployment of Decision Services, have additional information about the problem that is not displayed in the table. To view this information, hover over a *Failed* log message, and then click on the information button at the end of that line. An alert opens with additional information on the issue.

You can filter the table to view a subset of the log messages. To do this, select the filters you want from the drop-down lists, and then click **Filter**. For example, to view all failed Decision Services deployments by a user, select the username from the **User** drop-down, select **Decision Service** in **Component**, select **Deploy** in **Action** and finally, select **Failed** in the **Status** drop-down. You can also add a date range to the filter to narrow the information to only log messages recorded between specified dates.



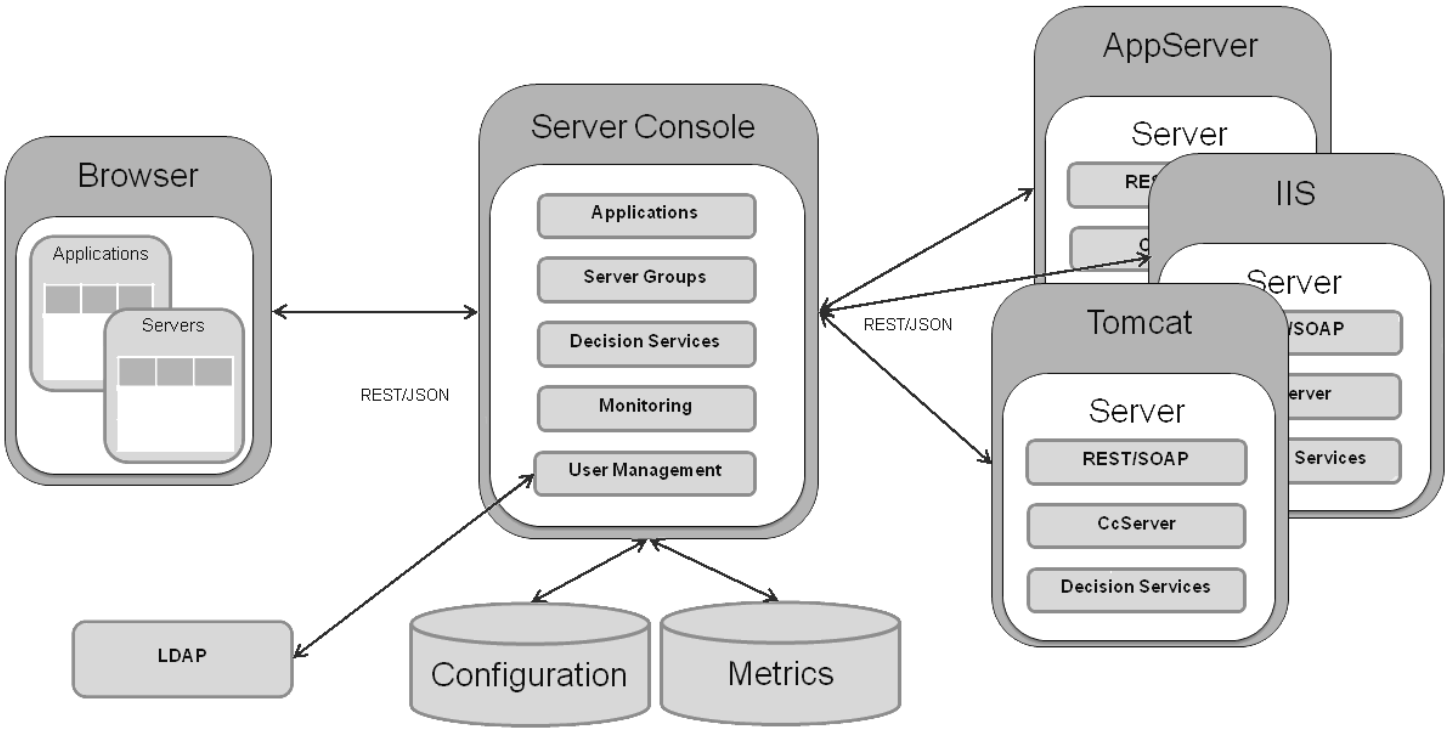
The Web Console maintains this log for a configurable period of time. This setting is visible only to Web Console Administrators. To know more about configuring the Activity Log, see the topic Configure the Activity Log on page 47.

3

Administrator Guide

Architecture Overview - The Web Console is a separate web application (`corticon.war`) from the Corticon Server (`axis.war`), deployable to either the same or separate application server as the Corticon Server.

When managing a group of Corticon Servers the recommended practice is to deploy the Web Console to a separate application server as depicted in this diagram:



Key aspects of this diagram:

- There is a single application server hosting the Web Console and three application servers hosting Corticon Servers. The Web Console is agnostic to the application server hosting a Corticon Server, this includes a mix of Java and Corticon Server for .NETs.
- REST/JSON is used for communication between the browser and the Web Console and between the Web Console and the Corticon Server.
- The Web Console stores all configurations locally. This includes definition of server groups, applications, and Decision Services (including the EDS file).
- The Web Console stores historical metrics locally. A retention policy will be supported for determining how long to keep historical metrics.

Installation

The Corticon Java and Corticon Server for .NET installers are in one installer, together with the Web Console. See the *Corticon Installation Guide* for more information. See the Progress Software web page Corticon Supported Platforms Matrix to review the currently supported browsers, platforms and application servers.

For details, see the following topics:

- User management
- Configure the Activity Log
- Configure auto logout
- Reset the administrator password

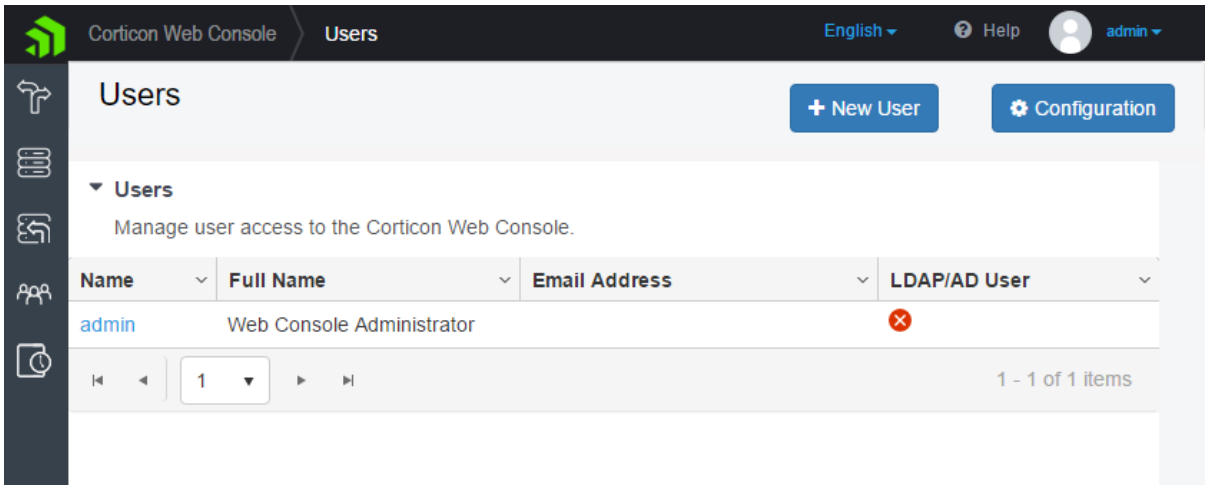
User management

The Web Console provides secure access. The administrator (User Name `admin`) is a preset user that cannot be deleted. You can change the administrator's password -- that's a task you should do as soon as you get started with the Web Console and take the administrator's role.

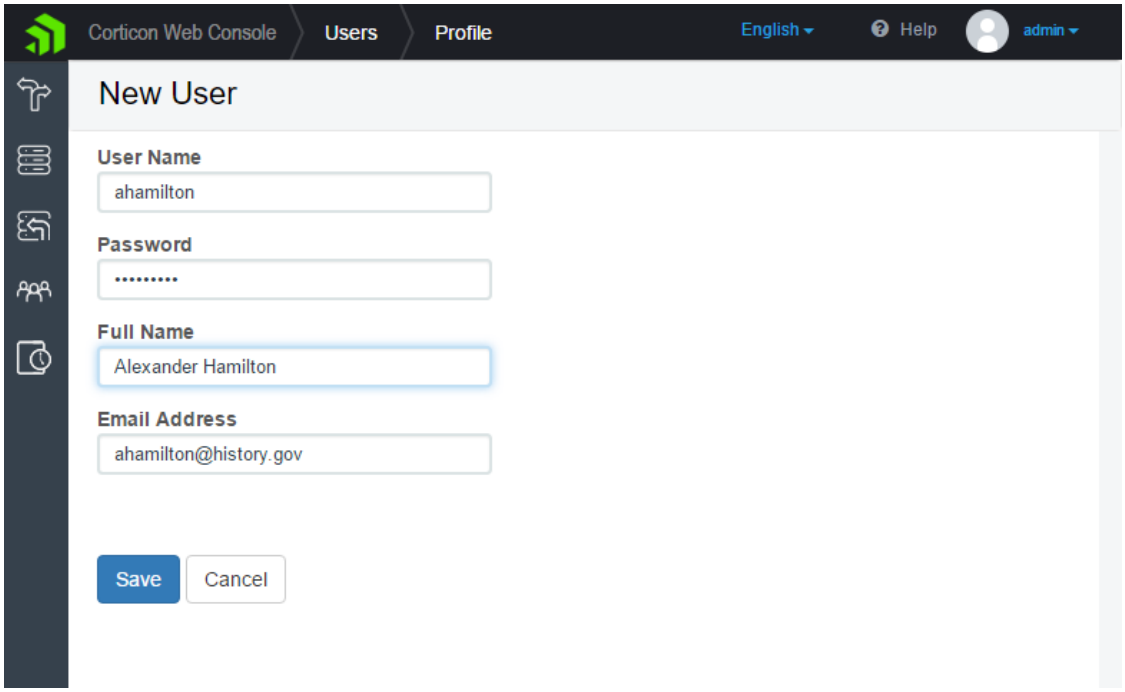
The administrator is the only user that can access user management to create, edit, and delete other users. Note that the case matters in the user name and password.

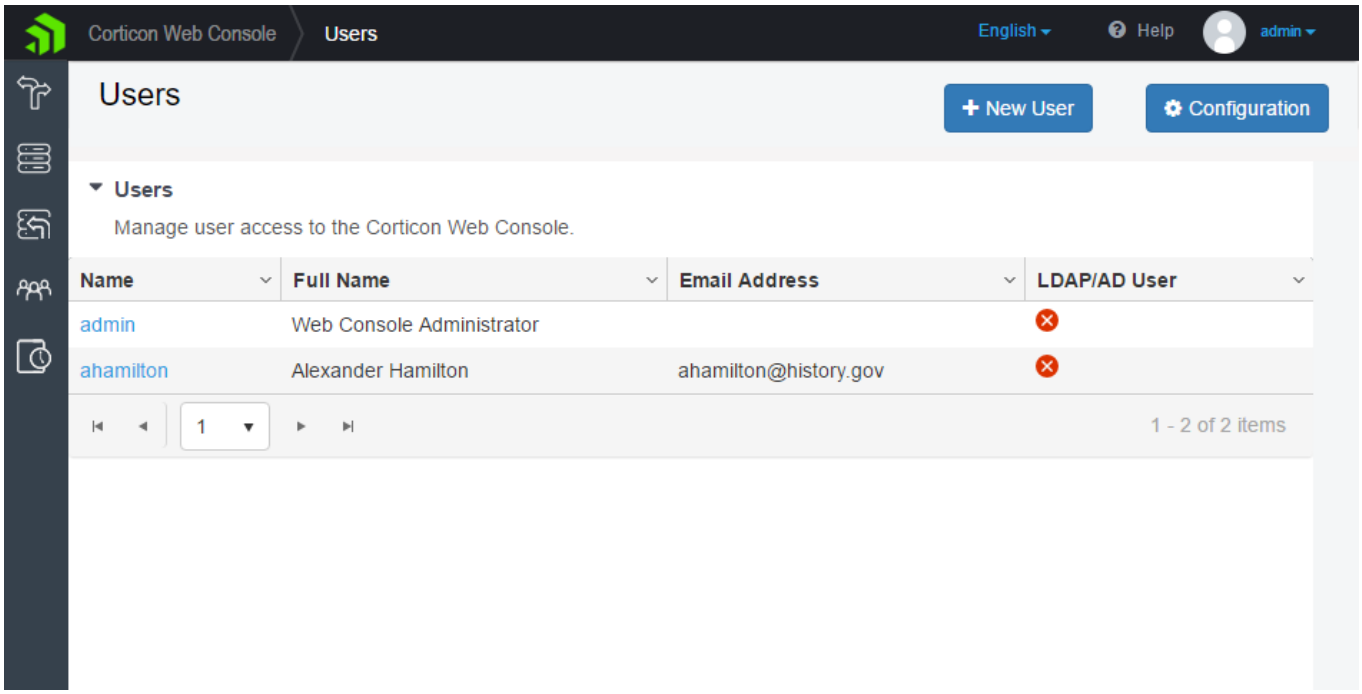
To display users:

1. Connect to the Web Console server as `admin` where you want to manage users.
2. Click the **Users** button:  The Users page opens:



To create new users, click **+ New User**, and then enter the user information and click **Save**:





How to use LDAP for Web Console authentication

You can also set up LDAP authentication, if business needs require your users to be authenticated through an LDAP server. After LDAP authentication is set up, LDAP users who log in to the Web Console are added to the **Users** page. LDAP users are differentiated from other users by the **LDAP/AD** annotation. Note that while LDAP users can be deleted from the **Users** page, their details cannot be modified in the Web Console.

To configure LDAP authentication, edit the file `CorticonServerConsoleConfig.groovy` located in `[CORTICON_WORK_DIR]\etc`.

Uncommment all property lines in this file and enter values for the first four properties. Here is an example:

```
/*Configure the manager distinguished name based on your ldap configuration*/
grails.plugin.springsecurity.ldap.context.managerDn = 'uid=admin,ou=system'

/*Ldap password*/
grails.plugin.springsecurity.ldap.context.managerPassword = 'secret'

/*Ldap server url */
grails.plugin.springsecurity.ldap.context.server = 'ldap://localhost:10389'

/*Ldap search base*/
grails.plugin.springsecurity.ldap.search.base = 'o=mojo,ou=groups'
```

To map Web Console Admin and User roles to LDAP user groups, specify the user group names in the `ldap{}`section at the bottom as shown. Use commas to define multiple user groups for each role.

```
corticon
{
    ldap {
        admin = "Admin,DBAdmin"
        users = "User,Analyst,Developer"
        authenticateUsersWithGroups = "false" //This flag is used to
authenticate the Admin and User groups listed above. If the flag is
false, and the group is not listed above, the user will be defaulted
to user role.

   }
}
```

After setting these properties, save the file and restart Corticon Server. LDAP users can log then in to Web Console using their LDAP user credentials. Once an LDAP user logs in, they are added to the **USERS** page in Web Console.

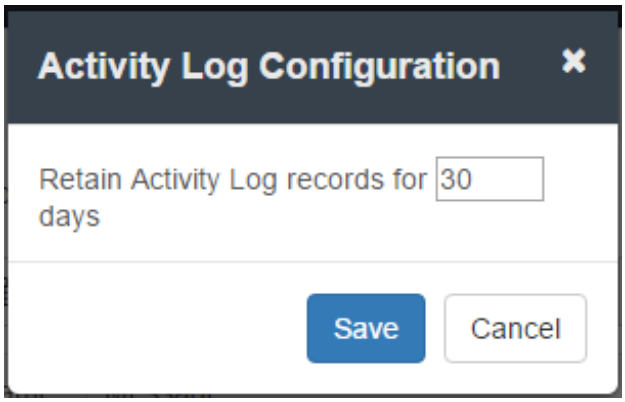
Note: Setting up LDAP authentication adds LDAP users to the Web Console user base. You can add other users in the **USERS** page and have them access Web Console using their Web Console user credentials.

Configure the Activity Log

Corticon Web Console maintains a log of its activities. The log includes:

- User actions such as deploying or undeploying Decision Services and creating or modifying Applications and Servers.
- System events such as deployment failures and lost connections to Servers.

A Web Console Administrator can view the Activity Log as well as configure the duration for which Corticon Web Console maintains log records. To view the Activity Log, click **ACTIVITY LOG** on the left pane. To configure the duration for which Web Console keeps log records, click **Configuration** on the Activity Log page and set the number of days for which Web Console maintains log records.



To know more about viewing and filtering the Activity Log, see the topic How to view the Activity Log on page 41.

Note: The Web Console Activity Log is different from a Server log, which logs user actions, system events, and other information for a specific instance of Corticon Server based on configurable log levels.

Update Loglevel for Web Console

The Corticon Web Console uses logback for logging. The `logback.xml` is located in the `Progress/Corticon_Server_Work_x.x/etc/` folder..

The log level is set to error by default. To enable info or debug level logging on the Web Console change the level to info or debug.

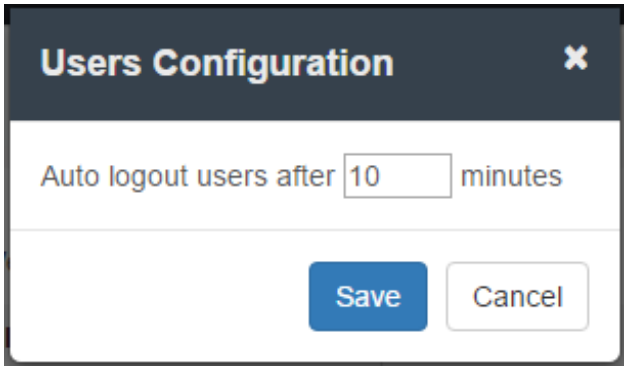
```
<logger name="com.progress.corticon.serverconsole"
       level="error"
       additivity="false">
    <appender-ref ref="FILE" />
</logger>
```

The Web Console logging supports all the various configurations supported by Logback framework.

You can easily customize the logging to add different appenders by updating the `logback.xml` file. Refer to the Logback documentation for more details.

Configure auto logout

As part of user management, you can define a period of inactivity (in minutes) after which a user is automatically logged out of the Web Console. To configure this setting, click **Configuration** on the **Users** page. In the **User Configuration** dialog box, enter the duration of inactivity, as shown below:



Reset the administrator password

If the login password of the Web Console administrative user (`admin`) is lost, Corticon provides a way to reset the password to the default (also `admin`).

To reset the administrator's password:

1. Stop the Corticon Server that is running the Web Console.
2. Select **Start > Progress > Corticon Command Prompt**.
3. Enter `set JAVA_OPTS=-DCORTICON_RESET_ADMIN_PASSWORD=true`.
4. Enter

```
Progress\Corticon 7.0\Server\bin\startServer.bat
```

Corticon Server starts and resets the administrator's password.

After completing these steps, you can connect the Web Console and log in with the default administrator credentials, user `admin`, password `admin`. It is good idea to immediately replace the default password with your preferred administrator password.

This procedure applies to the application server that is installed by Corticon Server, Apache Tomcat. You can perform similar steps for other supported application servers and platforms. Consult your application server documentation for how to pass the JVM system property `CORTICON_RESET_ADMIN_PASSWORD` to the server.

Note: Do not set this property in startup scripts as it will reset the password on each startup. This should be only done only when the password needs to be reset. Subsequent launches of Corticon Server and the Web Console should use the normal startup procedures.
